# 11

# *Genealogies and Citations*

## 11.1 Introduction

Time is responsible for a special kind of asymmetry in social relations, because it orders events and generations in an irreversible way. Social identity and position is partially founded on common ancestors, whether in a biological sense (birth) or in an intellectual manner: citations by scientists or references to predecessors by artists. This is social cohesion by common descent, which is slightly different from cohesion by direct ties (see Part II). Social communities and intellectual traditions can be defined by a common set of ancestors, by structural relinking (families that intermarry repeatedly), or by long-lasting cocitation of papers.

Pedigree is also important for the retrospective attribution of prestige to ancestors. For example, in citation analysis the number of descendants (citations) is used to assign importance and influence to precursors. Genealogy is the basic frame of reference here, so we discuss the analysis of genealogies first.

## 11.2 Example I: Genealogy of the Ragusan Nobility

Ragusa, which is now known as Dubrovnik (Croatia), was settled on the coast of the Adriatic Sea in the seventh century. For a time, it was under Byzantine protection, becoming a free commune as early as the twelfth century. Napoleon, having destroyed the Venetian Republic in 1797, put an end to the Republic of Ragusa in 1806. It came under Austrian control until the fall of the Austro-Hungarian monarchy in 1918.

In Ragusa, all political power was in the hands of male nobles older than eighteen years. They were members of the Great Council (*Consilium majus*) who had the legislative authority. Every year, eleven members of the Small Council (*Consilium minus*) were elected. Together with a

duke, the Small Council had both executive and representative authority. The main power was in the hands of the Senate (*Consilium rogatorum*), which contained forty-five members elected for one year. This organization prevented any single family, such as the Medici family in Florence, from prevailing. Nevertheless historians agree that the Sorgo family was among the most influential.

The Ragusan nobility evolved from the twelfth century to the fourteenth century and was finally established by statute in 1332. After 1332, no new families were accepted until the large earthquake in 1667. A major problem facing the Ragusan noble families was that, because of their decreasing numbers and the lack of noble families in the neighboring areas, which were under Turkish control, they became more and more closely related – marriages between third and fourth removed relatives were frequent. It is interesting to analyze how families of a privileged social class organized their relations by marriage and how they coped with the limited number of potential spouses for their children.

The file `Ragusan.ged` contains the members of the Ragusan nobility from the twelfth to the sixteenth centuries, their kinship relations (parent–child); their marriages; and their (known) years of birth, marriage, and death. Note that this is not an ordinary network file, because it contains attributes and ties of vertices. The extension `.ged` indicates that it is a GEDCOM file, which is the standard format for genealogical data, as explained in the next section. The genealogy is large; it contains 5,999 persons. For illustrative purposes, we selected the descendants of one nobleman, Petrus Gondola, in the file `Gondola_Petrus.ged` (336 persons).

## 11.3 Family Trees

Across the world, many people are assembling their family trees. They visit archives to collect information about their ancestors in registers of births, deaths, and marriages. Because in most Western societies family names are the usual entries in these registers and family names are the father's surname, a patrilineal genealogy is reconstructed, in which father–child relations rather than mother–child relations connect generations. In addition, marriages are included in the family tree.

Figure 95 shows a part of the Gondola family tree, which includes three generations of descendants of Petrus Gondola, who was born in 1356. Note that children born to a Gondola father are included because they receive the Gondola surname. Children of a Gondola mother are not included because their surname assigns them to another family in this historiography of a family name. An exception would be a Gondola mother

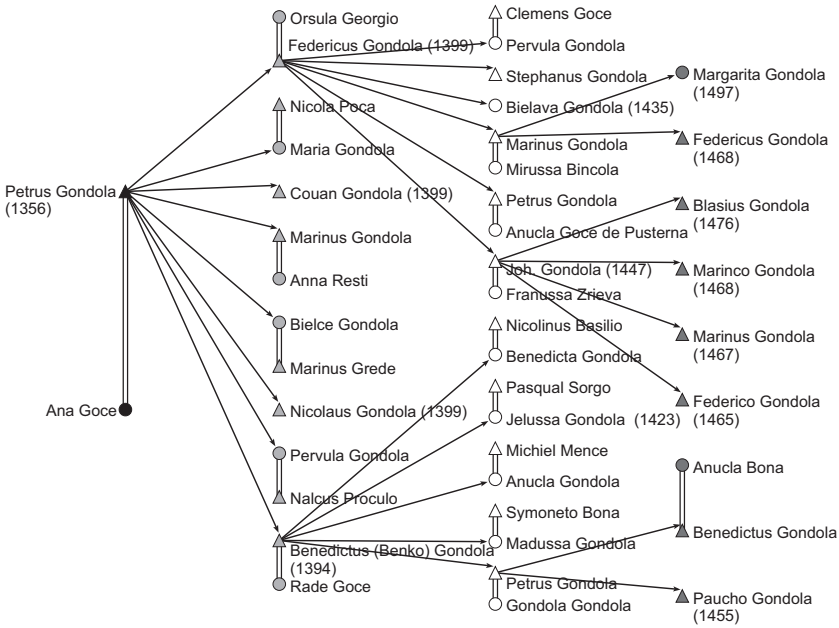



Figure 95. Three generations of descendants to Petrus Gondola (years of birth).

who married a Gondola father, but, as shown in Figure 95, this does not occur among the descendants.

In principle, genealogies contain persons as units and two types of relations among persons: birth and marriage. A person may belong to two nuclear families: a family in which he or she is a child and a family in which it is a parent. The former is called the *family of child or orientation* (FAMC) and the latter is the *family of spouse or procreation* (FAMS). Petrus Gondola's family of procreation, for example, contains his wife and eight children, and it is identical to the family of orientation of each of his children. A husband and wife have the same family of procreation, but they have different families of orientation unless they are brother and sister.

The standard data format for genealogies (GEDCOM) uses the double coding according to family of orientation and family of procreation. In addition, it has facilities to store all sorts of information about the persons and events (e.g., about their marriage), so we advise using this data format for the collection and storage of genealogical data. On the Internet, excellent free software and several databases of genealogical data are available (see "Further Reading").

In a representation of a genealogy as a network, family codes are translated to arcs between parents and children. In a sociogram of kinship ties

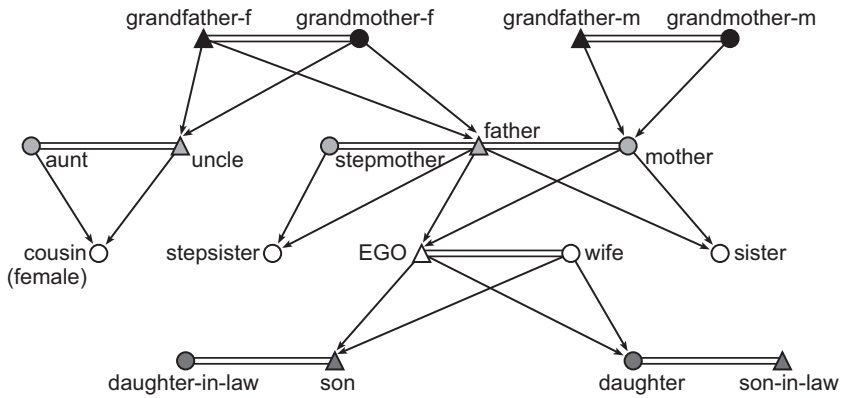



Figure 96. Ore graph.

that is known as the *Ore graph* (Figure 96), men are represented by triangles, women by ellipses, marriages by (double) lines, and parent–child ties by arcs. Note that the arcs point from parent to child following the flow of time.

In contrast to the family tree, fathers and mothers are connected to their children in an Ore graph. This greatly simplifies the calculation of kinship relations because the length and the direction of the shortest semipath between two individuals define their kinship tie; for instance, my grandparents are the vertices two steps up from me in the Ore graph. They are relatives in the second remove because two births are included in this path. In a patrilineal family tree, relatives from my mother's side (e.g., her parents and brother) are not included, so it is impossible to establish my kinship tie with them. In the Ore graph, it is possible to distinguish between blood relations and marriage relations, so we may calculate the remove in a strict sense, that is, ignoring marital relations, or in a loose sense, including them and considering them relations with zero distance.

In the standard display of a kinship network, marriages and siblings are drawn at the same layer, and layers are either top-down (Figure 96) or ordered from left to right (Figure 95). A layer contains a *genealogical generation*: grandparents versus parents, uncles and aunts versus children, nieces, and nephews. Such are the generations that we experience during our lives. From a social point of view, however, we define generations as birth cohorts (e.g., the generation of 1945–60). In contemporary Western societies, *social generations* contain people who were born within a period of approximately fifteen years. Genealogical generations overlap with social generations to a limited extent. For four or more generations, genealogical generations may group people of very different ages as a result of early marriage and childbearing in one branch of the family and late marriage in another branch. The birth years of the

great-grandchildren of Petrus Gondola, for instance, range from 1455 (Paucho) to 1497 (Margarita; see Figure 95). Biologically, the former could have been the latter's grandfather. As a consequence, Paucho's grandson could have married Margarita, causing a *generation jump* in the genealogy because it would connect a third-degree descendant of Petrus Gondola (viz., Margarita) to a fifth-degree descendant (Paucho's grandson).

The Ore graph is a very useful instrument for finding an individual's *ancestors* (*pedigree*) and *descendants* from both the father's and the mother's side. In addition, it is easy to count *siblings* and trace the *closest common ancestor* of two individuals.

*Application*

Genealogical data in GEDCOM format can be read directly by Pajek. To obtain the Ore graph, make sure that the option GEDCOM – *Pgraph in the Options> Read – Write* submenu is *not* selected before you open the GEDCOM file. When you check the option *Ore: Different relations for male and female links*, marriages receive line value and relation number 3 (drawn as double lines), father–child ties have a line value and relation number 1 (solid lines), and mother–child ties have a value and relation number 2 (dotted lines). This is particularly useful if you want to extract patrilineal ties from the Ore graph. In all other cases, it is better not to check this option, so all parent–child ties have line value and relation number 1. Then, open a GEDCOM file in the usual way with the *File> Network> Read* command, but select the option *Gedcom files* (*.ged) in the *File Type* drop-down menu of the Read dialog screen.

*[Main] Options> Read – Write> GEDCOM – Pgraph, Ore: Different relations for male and female links*

Reading the GEDCOM file, Pajek translates family numbers to parent–child ties and it creates two partitions and four vectors. The first obtained partition identifies vertices that are brothers and sisters, that is, children born to the same father and mother. Stepbrothers and stepsisters from a parent's remarriage are grouped separately, and vertices without parents in the network are collected in class 0. The second partition is the gender partition (class 1 for men, 2 for women). The vectors contain each person's sequential number in the GEDCOM file and his or her year of birth, marriage, and death. Unknown dates are represented by vector value 999999998. You may inspect the dates with the *Vector> Info* procedure in the usual way (see Section 2.5 in Chapter 2).

*Vector> Info*

The *genealogical generations* of the Ore graph can be obtained with the command *Genealogical* from the *Network> Acyclic Network> Create Partition> Depth Partition* submenu. An acyclic depth partition is not possible because the marriage edges are cyclic: A husband is married to his wife, and a wife is married to her husband at the same time. Draw the network in layers according to the genealogical depth partition (*Layers> In y Direction* in the Draw screen) and optimize it in the usual way (*Layers>*

*Network> Acyclic Network> Create Partition> Depth Partition> Genealogical*

*Layers> In y Direction*

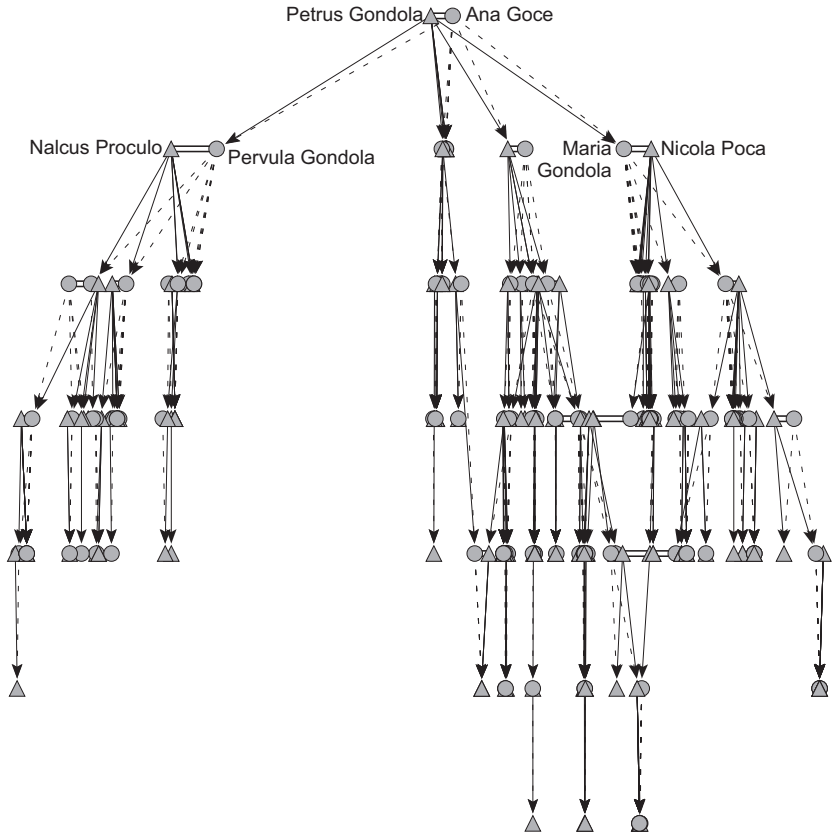



Figure 97. Descendants of Petrus Gondola and Ana Goce.

*Layers> Optimize Layers in x Direction*

*Layers> Averaging x Coordinate*

*[Main] Options> Read – Write> Ore: Different relations for male and female links*

*Network> Create New Network> Transform> Line Values*

*Optimize Layers in x Direction*). To focus on the distinct branches in the genealogy rather than the vertices, use the *Averaging x Coordinate* command from the *Layers* menu. Usually, the *Forward* option works well, but you may have to apply it more than once to clearly separate distinct branches as in Figure 97.

The length of the shortest semipath in a symmetrized Ore graph is the *remove* or *degree of a family relation*, provided that all parent–child ties have a line value of 1 and marriage lines have a line value of 0. Therefore, you must open the GEDCOM file with the option *Ore: Different relations for male and female links* not checked in the *Option> Read – Write* submenu. Marriage lines have value 3 (not 0 as in older versions of Pajek), so you must replace all line values 3 with value 0 before you calculate the degree of a family relation. You can do that in three steps all involving menu *Network> Create New Network> Transform> Line Values*. First use the *Add Constant* option and enter –3. In this way marriage links get

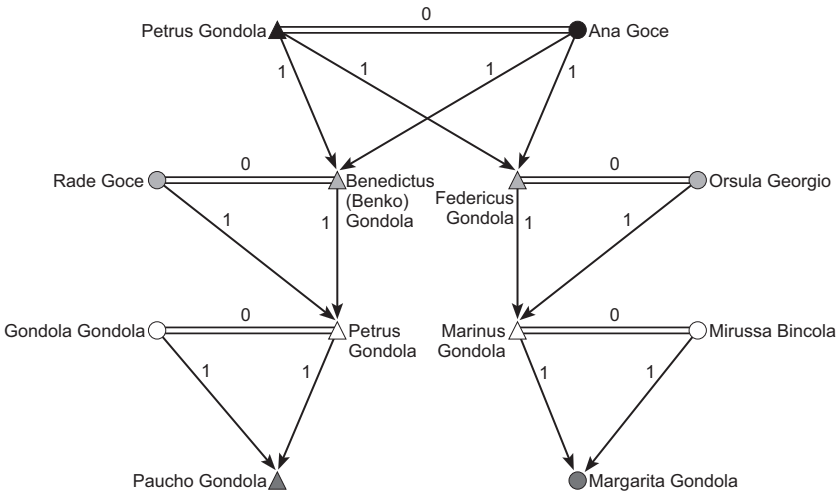



Figure 98. Shortest paths between Paucho and Margarita Gondola.

value 0, but now parent–child links have value –2; therefore, apply commands *Absolute* and subsequently *Multiply* by (0.5) to set all values of parent–child links back to 1 (marriage links stay 0).

First, decide whether you want to include marital relations in the calculation of the degree of family relations. If not, remove the edges from the network (*Network> Create New Network> Transform> Remove> all Edges*). Then, symmetrize the Ore graph (*Network> Create New Network> Transform> Arcs→Edges> All*; do not remove multiple lines), and use the *All Shortest Paths between Two Vertices* command to obtain the geodesics between two individuals in the network. When asked, do not ignore (forget) the values of the lines, because a marriage link should not contribute to the length of the semipath and hence to the remove of the relation. The length of the shortest paths, which is the distance between the vertices in the symmetrized network, is printed in the Report screen. Among the descendants of Petrus Gondola (Figure 95), for instance, Paucho Gondola (born in 1455) is a relative of Margarita Gondola (born in 1497) in the sixth remove.

*Network > Create New Network> Transform> Remove> all Edges*

*Network> Create New Network> Transform> Arcs→Edges> All*

*Network> Create New Network> SubNetwork with Path(s)> All Shortest Paths between Two Vertices*

Pajek creates a new network of the geodesics it has found and a partition that identifies the vertices on the geodesics in the original network provided that you requested this in one of the dialog boxes. If we extract these vertices from the original directed network (*Operations> Network + Partition> Extract> SubNetwork Induced by Union of Selected Vertices* and choose class 1) and relocate the vertices, we obtain the network shown in Figure 98. Note the triangles containing two

parents and one child. The direct path from child to father is just as long as the indirect path via the child's mother because a marriage line counts as 0 distance. If we had ignored line values, the shortest paths would not have included the mothers (except for Ana Goce) in this example.

In Figure 98, it is easy to see that Petrus Gondola and his wife Anna Goce are the closest common ancestors of Paucho and Margarita. Of course, we could already see that in the original family tree (Figure 95), but we need the shortest paths command in large networks such as the genealogy of the entire Ragusan nobility, because this is too complicated to analyze by eyeballing.

*Network> Create Partition> k-Neighbours*

The *ancestors* (*pedigree)* or *descendants* of a person are easily found with the *k-Neighbours* procedure in the Ore graph. Ancestors are connected by paths toward an individual, so they are its input neighbors. Descendants are reachable from the individual: They are output neighbors in the Ore graph. You may restrict the selection of ancestors to a limited number of generations in the *Maximum distance* dialog box of the *k-Neighbours* procedure. Note that the number of generations that you select is one more than the largest distance that you specify because the selected person, who also represents a generation, is placed in class 0. For example, the family tree in Figure 95 contains a number of output neighbors (descendants) of Petrus Gondola at maximum distance of 3.

*Partition> Info*

The Ore graph is most suited for finding brothers and sisters and for counting the size of sibling groups in a genealogical network. Pajek automatically creates a brothers/sisters partition, which identifies children of the same parental couple. Each class is a sibling group, except for class 0, so the number of vertices within a brothers and sisters class represents the size of a sibling group. Unfortunately, it is not easy to obtain a frequency distribution of the size of sibling groups from this partition in Pajek because the *Partition> Info* command lists each sibling group (class) separately.

It is possible, however, to obtain a frequency distribution of the size of sibling groups that have the same father or the same mother. In the Ore graph, the outdegree of a vertex is equal to the number of its children provided that marriage lines are disregarded. Ideally, every child has a father and a mother in the genealogical network, so we may count the number of children for each father or mother. In the case of a single marriage, the father and mother have the same number of children; but these numbers may differ in the case of remarriages. In the little example (Figure 96), my father remarried: He has three children (my stepsister, sister, and me), whereas my mother has only two children (my sister and me). Therefore, we must look at the outdegree of fathers or mothers, not at both.

Table 19. *Number of children of Petrus Gondola and his male descendants*

| Cluster | Freq | Freq% | CumFreq | CumFreq% | Representative |
|---|---|---|---|---|---|
| 0 | 131 | 67.5258 | 131 | 67.5258 | 4 |
| 1 | 14 | 7.2165 | 145 | 74.7423 | 15 |
| 2 | 15 | 7.7320 | 160 | 82.4742 | 3 |
| 3 | 11 | 5.6701 | 171 | 88.1443 | 1 |
| 4 | 7 | 3.6082 | 178 | 91.7526 | 2 |
| 5 | 4 | 2.0619 | 182 | 93.8144 | 29 |
| 6 | 1 | 0.5155 | 183 | 94.3299 | 120 |
| 7 | 3 | 1.5464 | 186 | 95.8763 | 23 |
| 8 | 4 | 2.0619 | 190 | 97.9381 | 13 |
| 9 | 1 | 0.5155 | 191 | 98.4536 | 114 |
| 11 | 2 | 1.0309 | 193 | 99.4845 | 85 |
| 12 | 1 | 0.5155 | 194 | 100.0000 | 171 |
| SUM | 194 | 100.0000 | | | |

This is achieved in the following way. First, remove the marriage lines (*Network> Create New Network> Transform> Remove> all Edges*) from the Ore graph. Now, the outdegree of a vertex is equal to an actor's number of children, so create an outdegree partition with the *Network> Create Partition> Degree> Output* command and select it as the first partition. Next we need the gender partition that was generated when reading the GEDCOM file as an Ore graph. In this partition, men are in class 1, women in 2. Select this partition as the second partition and execute the command *Partitions> Extract SubPartition (Second from First)*. In the dialog box, choose the class identifying the gender that you want to select, and Pajek will create a new partition with the outdegree of the selected vertices (e.g., the men).

*Network> Create New Network> Transform> Remove> all Edges*

*Network> Create Partition> Degree> Output*

*Partitions> Extract SubPartition (Second from First)*

The *Partition> Info* command will produce the desired frequency tabulation (see Table 19). Among Petrus Gondola's descendants, one man had twelve children and the others had fewer. Two-thirds (67.5 percent) of the men did not have children. Note, however, that they include the youngest men of the genealogy, who may have had children who were not included in the data set.

*Partition> Info*

From the parent–child and marriage relations in the Ore graph, several other types of family relation can be inferred. If, for example, we know that someone's child is a third person's parent, we know that the first person is a grandparent of the third person. We may create a network with arcs expressing grandparent relations if we want to analyze this type of family relation. With matrix multiplication, the grandparent and many other types of family ties can be created. Matrix multiplication is a standard operation in linear algebra, which requires two matrices and

produces a new matrix. We can conceptualize a network as a matrix (see Chapter 12), so we can apply this technique to networks.

*Network> Multiple Relations Network> Extract Relation(s) into Separate Networks*

*Networks> Multiply Networks*

*Network> Multiple Relations Network> Change Relation Number – Label*

*Network> Create New Network> Transform> 1-Mode to 2-Mode*

*Macro> Play*

The grandparent relation can easily be derived from an Ore graph. First, select the parent relation from the Ore graph (*Network> Multiple Relations Network> Extract Relation(s) into Separate Networks*). Second, multiply the parent's network by itself: select the parent's network in the first and second Network drop-down menus and issue the *Networks> Multiply Networks* command. The new network will contain the parents of parents of vertices: their grandparents. You may want to renumber and rename this relation with the *Network> Multiple Relations Network> Change Relation Number – Label* command (see earlier). Family relations networks are one-mode networks. It is also possible to multiply two-mode networks and one-mode with two-mode networks provided that the vertices of the one-mode network constitute one of the modes in the two-mode network and the one-mode network is changed into a two-mode network with the *Network> Create New Network> Transform> 1-Mode to 2-Mode* command.

In a similar way, many types of family relations can be established. Sometimes gender selections must be added – for example, if you want to tell grandfathers apart from grandmothers. The subdirectory `Macro>Kinship` in the directory where you installed Pajek contains macros for creating these networks. They require that you read the Ore graph with different relations for male and female links (see earlier). Use the *Macro> Play* command to select and execute these macros.

*Exercise I*

From the genealogical data in `Gondola_Petrus.ged`, construct a network containing Petrus Gondola (born in 1356) and all his descendants who received the Gondola surname at birth. In other words, create a patrilineal genealogy for Petrus Gondola's offspring.

## 11.4 Social Research on Genealogies

Kinship is a fundamental social relation that is extensively studied by anthropologists and historians. In contrast to people who assemble their private family trees, social scientists are primarily interested in the genealogies of entire communities, such as the nobility of Ragusa.

These genealogies, which are usually very large, enable the study of overall patterns of kinship ties that, for instance, reflect cultural norms for marriage: Who are allowed to marry? Property is handed over from one generation to the next along family lines, so marriages may serve to protect or enlarge the wealth of a family; family ties parallel economic exchange. Demographic data on birth, marriage, and death reflect

Table 20. *Size of sibling groups*[a] *in 1200–1250 and 1300–1350*

| | 1200–1250 | | 1300–1350 | |
|---|---|---|---|---|
| Size of Sibling Group | Freq | Freq% | Freq | Freq% |
| 0 (no children) | 18 | 16.4 | 386 | 54.5 |
| 1 | 22 | 20.0 | 87 | 12.3 |
| 2 | 18 | 16.4 | 73 | 10.3 |
| 3 | 19 | 17.3 | 53 | 7.5 |
| 4 | 11 | 10.0 | 38 | 5.4 |
| 5 | 10 | 9.1 | 29 | 4.1 |
| 6–10 | 12 | 10.9 | 40 | 5.6 |
| 11–21 | – | – | 2 | 0.3 |
| TOTAL (no. of sibling groups) | 110 | 100 | 708 | 100 |

[a] The number of children from one father.

economic and ecological conditions (e.g., a famine or deadly disease causes high mortality rates).

The number of marriages and the age of the marital couple and the size of sibling groups, nuclear families, or extended families are determined and compared across different societies or different periods. Differences are related to external conditions and internal systems of norms or rules.

Table 20 compares the number of children of Ragusan noblemen across two periods: men born in 1200–50 and 1300–50. Unfortunately, many birth dates are unknown, so we added the parents' children and the children's in-laws from the kinship network assuming that they belong to the same generation. In the Ore graph, the simple outdegree of a vertex specifies the number of children of a person. Table 20 summarizes the output degree frequencies. In the first half of the fourteenth century and in comparison to the previous century, a large proportion of the noblemen had no children. Perhaps fewer men got married because no new families were admitted to the nobility as of 1332. Conversely, some men may have died young as a consequence of the Black Death epidemic, which struck the town in 1348.

This type of research may use network analysis, but it can also be done by database counts, for instance, calculations on a GEDCOM genealogy database. A second type of research, however, is inherently relational and must use network analysis as a tool. It focuses on structural relinking between families and the economic, social, and cultural reasons or rules for structural relinking. *Structural relinking* refers to the phenomenon that families intermarry more than once in the course of time. Intermarriage or *endogamy* is an indicator of social cohesion within a genealogy. If families are linked by more kinship ties, they are more likely to act as a

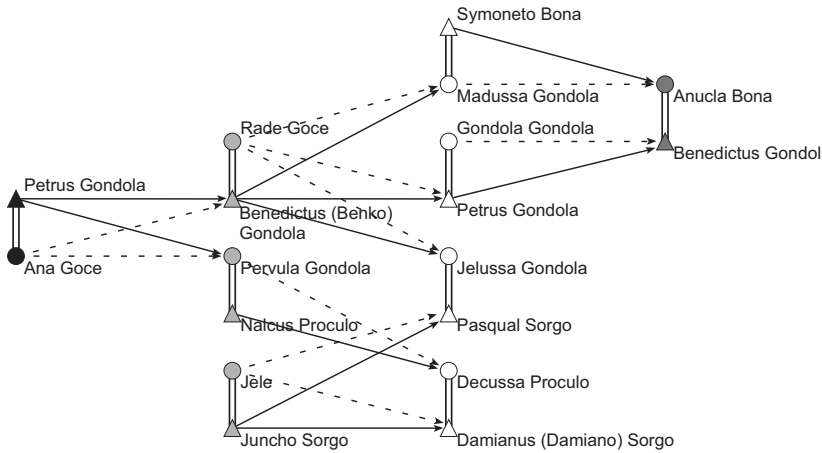



Figure 99. Structural relinking in an Ore graph.

clan: sharing cultural norms, entertaining tight relations, and restricting ties to families outside the clan.

There are two types of structural relinking: *blood marriages* and *nonblood relinking*. A blood marriage is the marriage of people with a close common ancestor, for instance, a marriage between a brother and sister or between a granddaughter and a grandson. The occurrence of this type of relinking tells us which types of intermarriages are culturally allowed and which are not. In the Ragusan nobility, a grandson of Benko Gondola (Benedictus Gondola) married a granddaughter (Anucla Bona), who was a fourth-degree relative (see Figure 99). Blood marriages between closer relatives – a son who married a daughter, a child who married a grandchild – did not occur among the Ragusan nobility. Apparently, these marriages were not allowed.

Nonblood relinking refers to multiple marriages between families without a close common ancestor. This type of relinking often serves economic goals, namely, to keep the wealth and power within selected families. Figure 99 shows nonblood interlinking between the Gondola and Sorgo families: two granddaughters of Petrus Gondola and Ana Goce (Jelussa and Decussa) marry brothers from the Sorgo family (Pasqual and Damianus), who were acknowledged to be the most influential family among the Ragusan nobility.

Structural relinking produces semicycles within a genealogical network; for instance, the blood marriage between Benedictus Gondola and Anucla Bona closes the paths from Benko Gondola to his granddaughter Anucla and his grandson Benedictus (Figure 99). The nonblood relinking between the Gondola and Sorgo families also yields a semicycle

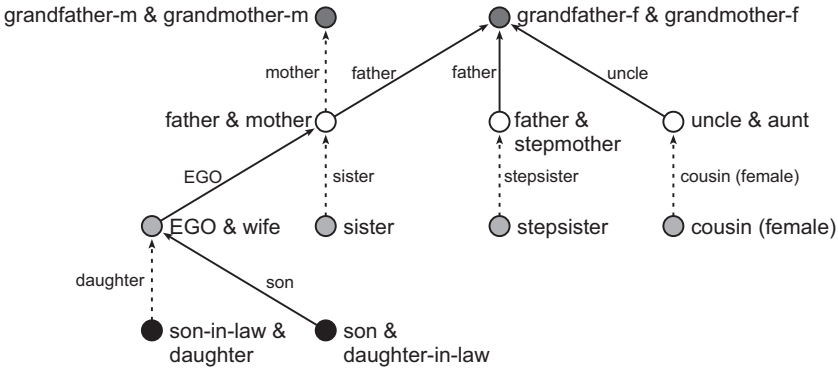



Figure 100. P-graph.

(Petrus Gondola–Benko–Jelussa Gondola–Pasqual Sorgo–Jele–Damianus Sorgo–Decussa Proculo–Pervula Gondola–Petrus Gondola, among other-semicycles).

However, in the Ore graph not all semicycles represent structural relinking. A father, mother, and child also create a semicycle (e.g., Ana Goce–Petrus Gondola–Pervula Gondola in Figure 99). In addition, parents and two or more children create larger semicycles (e.g., Ana Goce–Pervula Gondola Petrus Gondola–Benko Gondola–Ana Goce). Remarriages yield even more complicated semicycles that do not point to structural relinking.

Because it is troublesome to distinguish between semicycles that represent structural relinking and semicycles that do not, a special kind of genealogical network was developed: the *parentage graph* or *P-graph*. In the P-graph, couples and unmarried individuals are the vertices and arcs point from children to parents. The type of arc shows whether the descendant is male (full arc) or female (dotted arc). In Figure 100, for instance, my son and his wife are connected by a full arc to me and my spouse; my daughter and her husband are connected by a dotted arc.

The P-graph has several advantages. It contains fewer vertices, but the path distance in a symmetrized P-graph still shows the remove of a relation, although it is not possible to exclude marital ties from the calculation. The main advantage of the P-graph, however, is the fact that it is acyclic – there are no edges between married people – and there are no separate arcs from mother and father to child. As a result, every semicycle and bi-component indicates structural relinking, which is either a blood marriage or another type of relinking. Figure 101 shows the P-graph associated with the Ore graph of Figure 99. The two semicycles represent structural relinking: the blood marriage of Benedictus Gondola

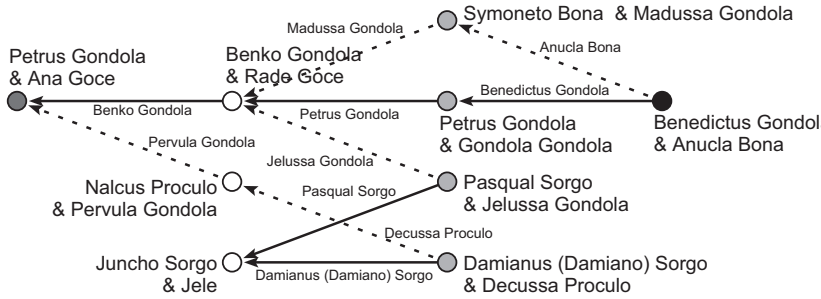



Figure 101. Structural relinking in a P-graph.

and Anucla Bona and the nonblood relinking between the Gondola and Sorgo families.

Apart from specific cases of relinking, social network analysts are interested in the amount of relinking in a genealogy. In a P-graph, this is measured by the *relinking index*. To understand this index, we must introduce the concept of a tree in graph theory: a connected graph that contains no semicycles. A tree has several interesting properties, but for our purposes the fact that it does not contain cycles and semicycles is most important.

A *tree* is a connected graph that contains no semicycles.

In a P-graph, every semicycle indicates structural relinking because the people or couples on the semicycle are linked by (at least) two chains of family ties (e.g., common grandparents on the father's side and on the mother's side). As a consequence, a P-graph that is a tree or a set of distinct trees (a *forest)* has no relinking, and its relinking index is 0. Given the number of people and the assumption that a marriage links exactly one man and one woman, the maximum amount of relinking within the P-graph of a genealogy can be computed so the actual number of relinking can be expressed as a proportion of this maximum. This is the relinking index, which is 1 in a genealogy with maximum relinking and 0 in a genealogy without relinking.

We advise calculating the relinking index on bi-components within the P-graph rather than on the entire P-graph. Genealogies have no natural borders; kinship ties extend beyond the boundaries of the data collected by the researcher, but boundary setting is important to the result of the relinking index. The largest bi-component within a genealogy is a sensible boundary because it demarcates families that are integrated into a system by at least one instance of relinking. In general, structural relinking may be used to bound the field of study, which means that you

limit your analyses to the families within the largest bi-component of a genealogy.

Let us calculate the amount of structural relinking among the Ragusan nobility in the period 1200–1350, in which new families were admitted to the nobility, and 1350–1500, when the nobility was chartered and no new families were admitted. Because we lack birth dates, we add the parents' children and children's in-laws to the couples in which at least one spouse is known to be born in the selected period. Between 1200 and 1350, a small number of the couples (137 of 1412 vertices, or 9.7 percent) were connected by two or more family ties, so the relinking index is low for the network in this period (0.02). Within this bi-component, the relinking index is higher (0.24), so there is a small core of families, the Sorgo family among them, who are tightly related by intermarriages.

In the period 1350–1500, the bi-component is larger, containing 476 couples (23.7 percent) and featuring many members of the Goce, Bodacia, and Sorgo families. The relinking index of the entire network is 0.20, and within the bi-component the proportion of relinking is 0.69. Both values are much larger than in the period before 1350, which shows increased endogamy among the Ragusan nobility.

In the P-graph, each person is represented by one arc except in the case of multiple marriages: remarriages and polygamy. Because each marriage is a separate vertex (e.g., my father and mother or my father and stepmother in Figure 100), men and women who remarry are represented by two or more arcs. In the P-graph, it is impossible to distinguish between a married uncle and a remarriage of a father or between stepsisters and (female) cousins. This problem is solved in the bipartite P-graph, which has vertices for individuals and vertices for married couples. However, the bipartite P-graph has the drawback of containing considerably more vertices and lines than the P-graph, and path distance does not correspond to the remove of a kinship relation. We do not use bipartite P-graphs in this book.

*Application*

The format of a genealogy that is read from a GEDCOM data file depends on the options checked in the *Options> Read – Write* menu. As noted, Pajek transforms a GEDCOM data file into an Ore graph if the option *GEDCOM – Pgraph* is *not* checked. A regular P-graph is created if this option is checked but the option *Bipartite Pgraph* is not. If the option *Pgraph + labels* is also checked, the name of a person is used as the label of an arc. All P-graphs have line value and relation number 1 for male lines and value 2 for female lines.

*Options> Read – Write> GEDCOM – Pgraph*

*Options> Read – Write> Bipartite Pgraph*

Pajek does not create a brothers and sisters partition in conjunction with a P-graph because siblings can easily be identified as the input neighbors (remember: arcs point from children to parents!) of a vertex

*Options> Read – Write> Pgraph + labels*

representing a married couple or an unmarried mother or father. It stores the years of birth of men and women in separate vectors because a couple has two birth dates. This also applies to the years of death. In addition, Pajek lists the year of marriage (999999998 for unmarried individuals), the family of spouse number (FAMS) for each couple, the family of child number (FAMC), and the sequential number (INDI) for the men and women separately.

We advise opening the entire Ragusan nobility genealogy (`Ragusan.ged`) as a P-graph (check option *GEDCOM-Pgraph* in the *Options> Read – Write* submenu) and making sure that names are used as labels of the arcs (also check the option *Pgraph + labels*). Note that reading the arc labels takes more time and uses more computer memory, so you may want to omit them if your network is very large and you do not really need the labels.

*[Draw] Options> Mark Vertices Using> Mark Cluster Only*

*Cluster> Create Empty Cluster*

As can be seen in Figure 101, the labels of vertices can be very long in P-graphs. This can make the layout difficult to read. Instead of showing all labels, you can show labels of selected vertices by selecting the option [Draw] *Options> Mark Vertices Using> Mark Cluster Only*. The same result can be obtained by selecting the checkbox at the right of the *Cluster* drop-down menu in the Main window following by clicking the drawing pencil button at the right of the *Network* drop-down menu. While this option is in effect, the vertices listed in the current cluster (if any), are labeled in the Draw screen. So create a cluster from a partition (see Section 8.3) or create an empty cluster (command *Cluster> Create Empty Cluster*) and manually edit the cluster so it contains the vertices for which labels should be visible. Another possibility is to display labels in several lines: insert *\n* at the position where the newline should occur. For example, change the label "Petrus Gondola & Ana Goce" to "Petrus Gondola \nAna Goce." Vertex labels can be changed if you manually edit a partition (*File>Partition> View/Edit*) that belongs to the network.

*File> Partition> View/Edit*

*Network> Acyclic Network> Info*

The relinking index is calculated by the *Network> Acyclic Network> Info* command, and it is printed in the Report screen. Note that the index is valid only for P-graphs. On request, Pajek will compute it for any acyclic network, but then its value is meaningless. In the P-graph with the entire Ragusan nobility, the relinking index is 0.23.

*Network> Create New Network> with Bi-Connected Components stored as Relation Numbers*

*File> Hierarchy> View/Edit*

If you want to calculate the relinking index for the largest bi-component in this P-graph, you have to identify the bi-components and extract the largest bi-component first. The *Network> Create New Network> with Bi-Connected Components stored as Relation Numbers* command, introduced in Chapter 7, identifies the bi-components. Make sure that the minimum size of a bi-component is set to 3 in the dialog box issued by this command. Recall that in Pajek bi-components are stored as relation numbers. Because relation numbers are already used in a P-graph to distinguish between male and female lines, store relation numbers in a new

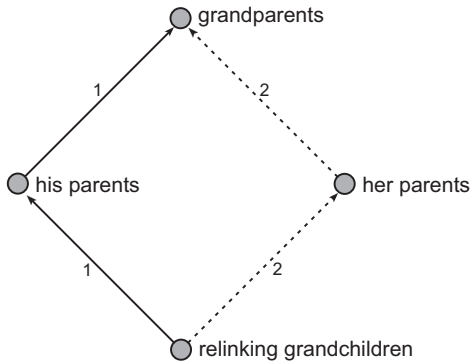



Figure 102. Fragment of relinking grandchildren.

network; do not overwrite old relation numbers because we may need them for other analyses. As you learned in Chapter 7, bi-components are stored also as a hierarchy, so inspect the hierarchy (*File> Hierarchy> View/Edit*) to find the sequential number and size of the largest bi-component. In the Ragusan nobility genealogy, we find two bi-components: the first contains 5 vertices and the second 1,446.

*Hierarchy> Extract Cluster*

*Operations> Network + Cluster> Extract SubNetwork*

*Network> Acyclic Network> Info*

Extract the second bi-component from the network in the following way: Translate the required class of the hierarchy into a cluster with the *Hierarchy> Extract Cluster* command, specifying the sequential number of the bi-component in the hierarchy, and execute the *Extract SubNetwork* command from the *Operations> Network + Cluster* menu. Finally, calculate the relinking index with the *Network> Acyclic Network> Info* command. The relinking index is 0.74, which is quite high. If you would like to draw this bi-component in layers, remember that the arcs point from children to parents in a P-graph, so the oldest generations are drawn at the bottom of the Draw screen.

*Networks> Fragment (First in Second)> Induced*

*Networks> Fragment (First in Second)> Find*

Particular types of relinking can be found with the *Fragment* commands in the *Networks* menu, which we also used to trace complete subnetworks (Chapter 3). Create a network that represents the relinking structure that you want to find (e.g., a marriage between two grandchildren of the same grandparents, see Figure 102), with the *Network> Create New Network> Empty Network* command and manual editing in the Draw screen. This fragment is also available in the file `relinking grandchildren.net`. Select this fragment as the first network, and select the P-graph of the Ragusan nobility genealogy as the second network. In the *Networks> Fragment (First in Second)* window, make sure that *Induced* is not checked because additional lines among the vertices in the fragment are allowed now. Finally, find the fragments with the *Find* command. Pajek encounters three instances of this fragment, among which is the marriage of the two grandchildren of Benko Gondola and Rade Goce.

*Networks> Fragment (First in Second)> Check values of lines*

*Networks> Fragment (First in Second)> Check relation numbers*

If you want to find a fragment with a particular pattern of male and female lines, make sure that the lines have the right values in the fragment (1 for male and 2 for female; the female lines do not have to be dotted) and select the *Check values of lines* option in the *Networks> Fragment (First in Second)* window. The same result can be obtained by matching the relation number (select the option *Check relation numbers* in the *Networks> Fragment (First in Second)* window). Recall that a line receives the same value as its relation number and line value when reading a GEDCOM file in Pajek, so you can use line values or relation numbers as a criterion for finding fragments (but do not forget to define line values and/or relation numbers also in the fragment.) In the Ragusan network, there are only two instances of a marriage among grandchildren of the same grandparents where the grandson is a descendant along patrilineal lines and the granddaughter descended along matrilineal lines as in the fragment of Figure 102.

When you want to restrict your analysis to a particular birth cohort, you need a network with a selection of the genealogical data. Because the vertices of a P-graph may represent couples, you have to take into account the years of birth of the men and women, which are stored in separate vectors. You may decide either that both husband and wife must be born in the selected period or that at least one of them must be in that period. We should note, however, that vertices may also represent unmarried individuals, in which case husband or wife is irrelevant. In addition, missing birth dates, which are to be expected in historical data, may cause problems if you demand that both husband and spouse are known to be born in the selected period. Given these complexities, we advise to select the right period in your genealogical database software, produce a separate GEDCOM data file, and have Pajek translate it into a P-graph. Then, skip the remainder of this section.

*Vector> Make Partition> by Intervals> Selected Thresholds*

If this is not possible, however, you may extract the subnetwork in Pajek by combining information from different vectors. First, translate the vectors with birth dates of men and women to partitions with the *Vector> Make Partition> by Intervals> Selected Thresholds* command. In the dialog box, enter the limits of the required period (e.g., 1349 and 1500 if you are interested in the people born in 1350 up to and including 1500). Note that each threshold is included as the upper limit of the interval. In addition, include the threshold 999999997 to obtain a separate class with the 999999998 code, which represents either unknown or irrelevant birth dates (e.g., the male birth date in the case of an unmarried woman). Separate the thresholds by a blank.

*Partition> Info*

If we execute the command, Pajek creates a partition with four classes. If we inspect the partition with male birth dates (*Partition> Info*), we see that 1,025 men were born before 1350, 1,493 were born between 1350 and 1500, and 46 were born after 1500, and we have no information on

Table 21. *Birth cohorts among men and women*

Rows: 10. From Vector 1 [1349 1500 999999997] (4376)
Columns: 11. From Vector 2 [1349 1500 999999997] (4376)
Crosstabs

| | 1 | 2 | 3 | 4 | Total |
|---|---|---|---|---|---|
| 1 | 51 | 1 | 0 | 973 | 1025 |
| 2 | 3 | 83 | 0 | 1407 | 1493 |
| 3 | 0 | 0 | 0 | 46 | 46 |
| 4 | 268 | 317 | 19 | 1208 | 1812 |
| TOTAL | 322 | 401 | 19 | 3634 | 4376 |

1,812 couples or individuals. The partition with female birth dates shows that 401 women are known to have been born between 1350 and 1500.

The four classes in the men and women partitions yield sixteen combinations, which are listed in Table 21. This table is part of the output produced by the *Partitions> Info> Cramer's V. Rajski, Adjusted Rand Index* command after selecting the male birth dates partition as the first partition and the female birth dates partition as the second. Note that the men are in the rows and the women in the columns and that the second class represents the period 1350–1500, whereas the fourth class contains the unknown and irrelevant birth dates.

*Partitions> Info> Cramer's V, Rajski, Adjusted Rand Index*

In Table 21, the second row contains the men who were born between 1350 and 1500 (1,493 in total), and the second column shows the (401) women born in this period. Only 83 couples are known to consist of a husband and wife born in the selected period. In a majority of cases, we deal with unmarried men or unknown birth date of the wife (1,407 cases) and unmarried women or unknown birth date of the husband (317 cases). In very few cases, one spouse is known to be born in the right period, whereas the other is born in another period, namely, before 1350 (period 1): In one case, the husband was born before 1350, and in three cases the wife was born before 1350.

It seems reasonable to select all vertices in which either the man or the woman was born in the right period. This can be done if we create a new partition identifying the vertices for which the male birth date and/or the female birth date is coded as class 2. First we have to binarize the two birth dates partitions such that the period 1350–1500 (class 2 in these partitions) becomes class 1 in the new partitions, whereas all other classes become 0. Simply execute the *Partition> Binarize Partition* command on each of the birth dates partitions and select class 2 in the dialog box. Do this for both partitions: male and female birth dates.

*Partition> Binarize Partition*

Then select the two binarized partitions as first and second partition and sum them (*Partitions> Add (First + Second)*). The resulting partition

*Partitions> Add (First + Second)*

has three classes: class 0 containing (2,565) individuals or couples without known birth between 1350 and 1500, class 1 containing (1,728) individuals and couples containing a husband or wife born in this period, and class 2 with (83) couples with both spouses known to be born between 1350 and 1500. Now we can extract the desired subnetwork from the Ragusan nobility genealogy by executing the *Operations> Network + Partition> Extract> SubNetwork Induced by Union of Selected Clusters* command, selecting clusters from one to two in the dialog box. This subnetwork contains 1,811 vertices.

*Operations> Network + Partition> Extract> SubNetwork Induced by Union of Selected Clusters*

*Macro> Play*

In the Ragusan nobility genealogy, many birth dates are missing. Assuming that all children of the same parents and all parents and in-laws of children belong approximately to the same birth cohort, we may add them to the people we know were born in the required period. We need these indirect neighbors to preserve the structure of the genealogical network. The procedure is stored in the macro `expand_generation.mcr`, which can be executed with the *Macro> Play* command. A genealogical network (Ore graph or P-graph) must be selected in the *Network* drop-down menu and a binary partition identifying the selected birth cohort must be selected in the *Partition* drop-down menu. Note that the partition that we used to extract the birth cohort is not binary, because it contains classes 0, 1, and 2. We must first binarize it such that all selected couples and individuals are in class 1. Execute the *Partition> Binarize Partition* command and select classes 1 and 2 in the dialog box if you want to expand this birth cohort. The macro creates a new partition with the extended birth cohort in class 1: in our example 2,007 bachelors and couples.

*Operations> Vector + Partition> Extract SubVector*

*Vector> Info*

The macro can be executed several times to increase the number of selected vertices, but *generation jumps* may extend the range of birth dates enormously. We advise applying the macro only once and checking the range of known birth years among the selected vertices afterward. To this end, extract the vertices selected in the expanded partition from the year of birth vector(s): Make sure the expanded birth cohort partition is selected in the *Partition* drop-down menu and a year of birth vector is selected in the *Vector* drop-down menu and execute the *Operations> Vector + Partition> Extract SubVector* command (select class 1 only). You may inspect the extracted years with the *Vector> Info* command, which reports the lowest and highest values: There should not be years that fall widely outside the selected period. In the case of a P-graph, you must check the birth dates of men and women separately. With the men, the known birth dates range from 1280, which is seventy years before the selected period, to 1500. The women were born between 1298 and 1498. Even in its first step, the expansion macro lengthens the range of birth dates considerably.

*Exercise II*
What kind of structural relinking does the small bi-component in the Ragusan nobility genealogy represent: a blood marriage or nonblood relinking? Extract this bi-component from the network, and draw it to find the answer to this question.

## 11.5 Example II: Citations among Papers on Network Centrality

In several social domains, genealogical terminology is used as a metaphor for nonbiological affinity. Artists who were trained by the same master or who are influenced by the same predecessors are considered to belong to the same family or tradition. A work of art has a pedigree: a list of former owners. In a similar way, scientists are classified according to their intellectual pedigree: the theories and theorists they use as a frame of reference in their work.

In science, citations make explicit this frame of reference, so they are a valuable source of data for the study of scientific development and scientific communities in scientometrics, history, and the sociology of science. They reveal the impact of articles and their authors on later scientific work, and they signal scientific communities or specialties that share knowledge.

In this chapter, we analyze the citations among articles that discuss the topic of network centrality. In 1979, Linton Freeman published an article that defined several kinds of centrality. His typology has become the standard for network analysis, so we used it in Chapter 6 of this book. Freeman, however, was not the first to publish on centrality in networks. His article is part of a discussion that dates back to the 1940s. The network depicted in Figure 103, (`centrality_literature.net`) shows the articles that discuss network centrality and their cross-references until 1979. Arcs represent citations; they point from the cited article to the citing article.

In principle, articles can cite only articles that appeared earlier, so the network is *acyclic*. Arcs never point back to older articles just as parents cannot be younger than their children. However, there are usually some exceptions in a citation network: articles that cite one another (e.g., articles appearing at about the same time and written by one author). We eliminate these exceptions by removing arcs that are going against time or by shrinking the articles by an author that are connected by cyclic citations. In the centrality literature network, we used the latter approach (e.g., two publications by Gilch denoted by #GilchSW-54 in Figure 103).

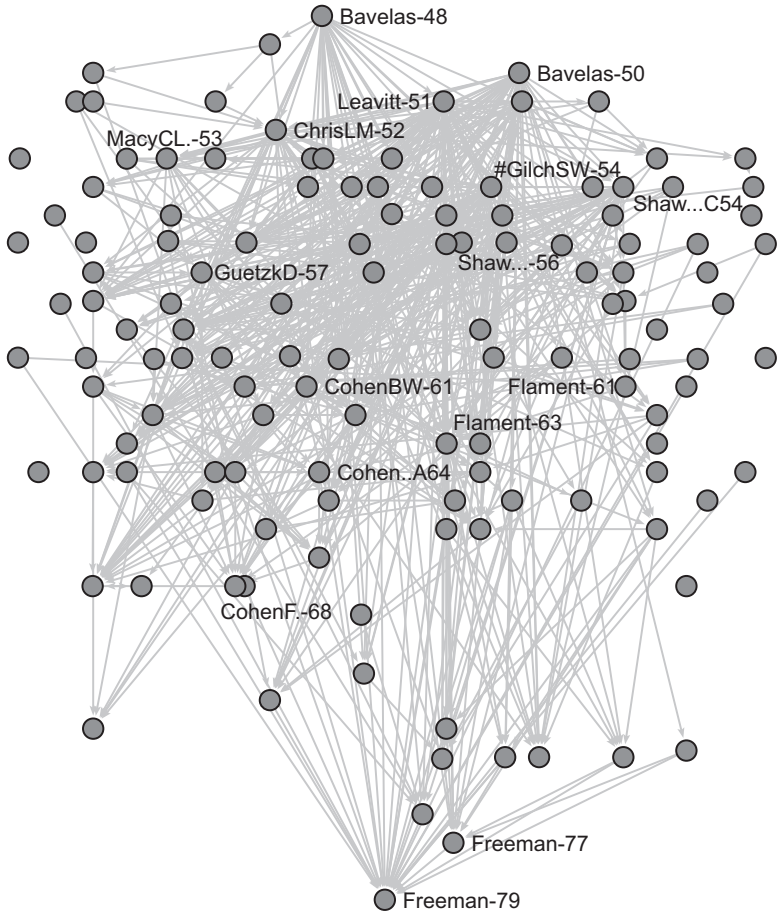



Figure 103. Centrality literature network in layers according to year of publication.

There are important differences between genealogical data and citation data. A citation network contains one relation, whereas genealogical data concern two relations: parenthood and marriage. In addition, an article may cite all previous articles notwithstanding their distance in time. In a genealogical network, children have two (biological) parents and parenthood ties always link two successive generations. The concept of a generation is not very useful in the context of a citation network, so we order the articles by publication date. In Figure 103, layers represent the year of publication (the `centrality_literature_year.clu` partition), which is also indicated by the last two digits in the label of a vertex.

## 11.6 Citations

Nowadays, citations are being used to assess the scientific importance of papers, authors, and journals. In general, an item receiving more citations is deemed more important. Bibliographic databases, for instance, the *Web of Science* compiled by the Institute for Scientific Information (ISI), list the citations in a large number of journals. Simple calculations yield indices of scientific standing, for instance, the *impact factor* of a journal (the average number of citations to papers in this journal) and the *immediacy index* (the average number of citations of the papers in a journal during the year of its publication). In each year, journals are ranked by their scores on these indices. Compared over longer periods, these indices show differences between scientific disciplines. In the liberal arts, for instance, it is rare for authors to cite recent publications, whereas this is very common in the natural sciences.

Citation analysis is not exclusively interested in the assessment of scientific standing. It also focuses on the identification of specialties, the evolution of research traditions, and changing paradigms. Researchers operating within a particular subject area or scientific specialty tend to cite each other and common precursors. Citation analysis reveals such cohesive subgroups, and it studies their institutional or paradigmatic background. Scientific knowledge is assumed to increment over time: Previous knowledge is used and expanded in new research projects. Articles that introduce important new insights are cited until new results modify or contradict them. Citation analysis, therefore, may spot the articles that influence the research for some time and link them into a research tradition that is the backbone of a specialty. Scientific revolutions, that is, sudden paradigmatic changes resulting from new insights, are reflected by abrupt changes in the citation network.

Network analysis is the preferred technique for extracting specialties and research traditions from citations. Basically, specialties are cohesive subgroups in the citation network, so they can be detected with the usual techniques. Weak components identify isolated scientific communities that are not aware of each other or who see no substantial overlap between their research domains. Within a weak component, a bi-component identifies sections where different lines of citations emanating from a common source text meet again. This is similar to the concept of relinking in genealogical research.

In most citation networks, however, these criteria are not strong enough because almost all articles are linked into one bi-component. $k$-Cores (Chapter 3) offer a more penetrating view. The centrality literature network, for example, contains one large weak component and eleven isolates. There is one large bi-component, and twelve vertices are connected by one citation. The network contains a 10-core of twenty-nine papers


Figure 104. *k*-Cores in the centrality literature network (without isolates).

that is the central summit of this network (the black vertices in Figure 104). Each of the articles in this core is connected to at least ten other articles by citations, but we do not know which are cited often and cite others often.

The cohesion concept (as discussed in Chapters 3–5) does not take time into account. It does not reflect the incremental development of knowledge, nor does it identify the articles that were vital to this development. Therefore, a special technique for citation analysis was developed that explicitly focuses on the flow of time. It is called *main path analysis*.

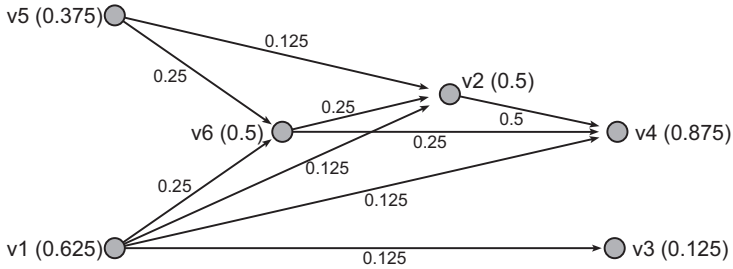



Figure 105. Traversal weights in a citation network.

Let us think of a citation network as a system of channels that transport scientific knowledge or information. An article that integrates information from several previous articles and adds substantial new knowledge receives many citations, and it will make citations to previous articles more or less redundant. As a consequence, it is an important junction of channels and a great deal of knowledge flows through it. If knowledge flows through citations, a citation that is needed in paths between many articles is more crucial than a citation that is hardly needed for linking articles. The most important citations constitute one or more main paths, which are likely to be the backbones of a research tradition.

Main path analysis calculates the extent to which a particular citation or article is needed for linking articles, which is called the traversal count or traversal weight of a citation or article. First, the procedure counts all paths from each source (an article that is not citing within the data set) to each sink (an article that is not cited within the data set), and it counts the number of paths that include a particular citation. Next, it divides the number of paths that use a citation by the total number of paths between source and sink vertices in the network. This proportion is the traversal weight of a citation. In a similar way, you can obtain the traversal weight of each article.

In an acyclic network, a *source vertex* is a vertex with 0 indegree.

In an acyclic network, a *sink vertex* is a vertex with 0 outdegree.

The *traversal weight* of an arc or vertex is the proportion of all paths between source and sink vertices that contain this arc or vertex.

For example, Figure 105 shows a citation network of six articles ordered in time from left to right. There are two sources (v1 and v5) and two sinks (v3 and v4). One path connects source v1 and sink v3, but

there is no path from v5 to v3. Four paths reach v4 from v1 and three paths from v5. In sum, there are eight paths from sources to sinks. The citation of article v1 by article v3 is included in one of the eight paths, so its traversal weight is 0.125. The citation of v2 in article v4 is contained in exactly half of all paths. The traversal weights of the vertices, which are reported between brackets, are calculated in a similar way.

Now that we have defined and calculated the traversal weights of citations, we may extract the paths or components with the highest traversal counts on the lines, the main paths or main path components, which are hypothesized to identify the main stream of a literature. We can analyze their evolution over time and search for patterns that reflect the integration, fragmentation, or specialization of a scientific community.

In a citation network, a *main path* is the path from a source vertex to a sink vertex with the highest traversal weights on its arcs. Several methods have been proposed to extract main paths from the network of traversal weights. The method we will explain first is called *forward local main path search*. It consists of choosing the source vertex (or vertices) incident with the arc(s) with the highest weight, selecting the arc(s) and the head(s) of the arc(s), and repeating this step until a sink vertex is reached. In the example of Figure 105, the main paths start with vertex v1 and vertex v5 because both source vertices are incident with an arc carrying a traversal weight of 0.25. Both arcs point toward vertex v6, which is the next vertex on the main paths. Then, the paths proceed either to vertex v2 and on to vertex v4 or directly from vertex v6 to vertex v4. We find several main paths, but they lead to the same sink, so we conclude that the network represents one research tradition.

Instead of starting with one or more source vertices, we may start with one or more sink vertices incident with the arc(s) with the highest weight and travel against the direction of the arcs. Thus, we get the *backward local main path search*. In the example of Figure 105, we find the backward local main paths starting with sink vertex v4, which is incident to an arc with traversal weight 0.5, because the other sink vertex (v3) is linked to an arc with a much lower traversal weight (only 0.125). From vertex v4 we proceed backwards to vertex v2, from there to vertex v6, finally reaching both vertex v1 and vertex v5.

In *key-route local main path search*, we select a limited number of arcs, for example, 10. These are usually the arcs with the highest traversal weights. Selected arcs are called *key-routes*. Note that key-routes do not need to be arcs attached to source or sink vertices. They can be located anywhere in the acyclic network, so this is a main difference with forward and backward local searches. For each key-route, we find the main path from source vertices to the key-route and from the key-route to sink

vertices. We search forward from the terminal vertex of the key-route until a sink vertex is reached. In each step we select the arc(s) with the highest traversal weight(s). Then we search backward from the initial vertex of the key-route until a source vertex is reached. The resulting *Key-route local main path* consists of the main paths obtained for all key-routes. If we run the key-route local main path search with the highest arc as the only key-route for the example network of Figure 105, we get the same result as with forward or backward local main path search.

The three methods defined so far are called *local main path methods* because we search only locally for the current arc in each step; that is, we just check arcs that are incident with the current vertex and have the right direction. In local main path searches, we may relax the rule that we select only the arc(s) with the highest value in each step. If we also accept arcs with weights slightly below the highest value, we introduce *tolerance* in the search. If tolerance is set to 0, we select only arcs with the highest value. But if tolerance is set to some positive value, for example, 0.1, and the highest arc value is 0.5, all arcs with values larger than 0.4 are selected. It is usually a good idea to start with zero tolerance to avoid finding very extensive ("broad") main paths.

In contrast to local search methods, *global main path methods* search for paths with the overall highest sum of traversal weights. Allowing for tolerance here is not a good idea because the search can become computationally demanding. We present two global methods: *Standard global main path search* and *Key-route global main path search*.

*Standard global main path* is the path from source to sink vertices with the overall highest sum of traversal weights on the path. This method is widely used across scientific disciplines. In project planning, for example, it is called *Critical Path Method* (*CPM*). CPM is an algorithm for scheduling a set of project activities. For the example in Figure 105, s*tandard global main path* yields the same result as the one obtained using forward or backward local main path search. The overall highest sum of traversal weights is 1.

In *Key-route global main path* search, we again start with some arcs as *key-routes*. For each key-route we search for the main path that contains the key-route from source to sink vertices with the overall highest traversal weight. The *Key-route global main path* unites the main paths obtained for all key-routes. In the example of Figure 105, key-route global main path search with the highest arc as the only key-route gives us the same result as forward and backward local main path search because the arc from vertex v2 to v4 is both the key-route (traversal weight is 0.5) and part of the forward and backward main paths.

Usually some arcs (citations) with the highest traversal weights are selected as key-routes. But this is not necessary. In citation network

analysis, it may make sense to select one or more papers (not citations) that are of particular interest, for example some papers that you wrote yourself, and search for the main path containing these papers. For more details on main path searches check the references in Further Reading.

A *main path component* is extracted in the following way. Choose a cutoff value between 0 and 1, and remove all arcs from the network with traversal weights below this value. The components in the extracted networks are called main path components. Usually, we look for the lowest cutoff value that yields a component that connects at least one source vertex to one sink vertex. This value is equal to the lowest traversal weight on the main paths. In our example, this cutoff value is 0.25, and we obtain a main path component that includes all articles except v3, which is a marginal article in the research tradition represented by this data set.

Of course, article v3 may be very important in another research tradition. The choice of the articles to be included in the data set restricts the number and size of research traditions that can be found. Like a genealogy, a citation network is virtually endless, so it cannot be captured entirely in a research project. The researcher has to set limits to the data collection, but this should be based on sound substantive arguments.

Citation networks are usually created from bibliographic databases such as the *Web of Science*. The bibliographic data stored in these databases also allow the creation of other types of networks: coauthorship networks, bibliographic coupling networks, cocitation networks, keyword networks, and so on. See the Further Reading section for references and for a link to software that transforms downloads from bibliographic databases into Pajek networks.

*Application*

In Chapters 3 and 7, we discussed the commands for detecting components, bi-components, and *k*-cores, which identify cohesive subgroups in a network. In principle, a citation network is directed and acyclic, so you should search weak components instead of strong components and find *k*-cores on input and output ties (command *All* in the *Network> Create Partition> k-Core* submenu).

*Network> Create Partition> k-Core> All*

Main path analysis is very easy in Pajek. The commands in the *Network> Acyclic Network> Create Weighted Network + Vector> Traversal Weights* submenu compute the traversal weights for lines and vertices in an acyclic network. There are three commands: *Search Path Count (SPC)*, *Search Path Link Count (SPLC)*, and *Search Path Node Pair (SPNP)*. The *Search Path Count (SPC)* command counts the paths between all source and sink vertices as explained previously. The *Search Path Link Count (SPLC)* command traces paths from all vertices to the sink vertices. In the latter procedure, citations of early articles receive lower weights because they cannot be part of paths emanating from later

*Network> Acyclic Network> Create Weighted Network + Vector> Traversal Weights*

Table 22. *Traversal weights in the centrality literature network*

| Line Values | Frequency | Freq% | CumFreq | CumFreq% |
|---|---|---|---|---|
| (.... 0.0000] | 90 | 14.68 | 90 | 14.68 |
| (0.0000 .... 0.0515] | 465 | 75.86 | 555 | 90.54 |
| (0.0515 .... 0.1030] | 45 | 7.34 | 600 | 97.88 |
| (0.1030 .... 0.1545] | 8 | 1.31 | 608 | 99.18 |
| (0.1545 .... 0.2059] | 2 | 0.33 | 610 | 99.51 |
| (0.2059 .... 0.2574] | 2 | 0.33 | 612 | 99.84 |
| (0.2574 .... 0.3089] | 0 | 0.00 | 612 | 99.84 |
| (0.3089 .... 0.3604] | 0 | 0.00 | 612 | 99.84 |
| (0.3604 .... 0.4118] | 1 | 0.16 | 613 | 100.00 |
| TOTAL | 613 | 100.00 | | |

articles, so we advise to use it only in special cases where early articles are relatively unimportant. In the *Search Path Node Pair (SPNP)* command, each vertex is considered as a source and as a sink. As a result, vertices and edges in the middle will receive higher traversal weights.

*Network> Acyclic Network> Create Weighted Network + Vector> Traversal Weights> Normalization of Weights*

There are several ways of normalizing the traversal weights of lines and vertices in a citation network. Previously we discussed the normalization according to flow (*Network> Acyclic Network> Create Weighted Network + Vector> Traversal Weights> Normalization of Weights> Normalize–Flow)* for the *Search Path Count* method: the number of paths that include a line or vertex divided by the total number of paths between sources and sinks. This normalization yields the percentage of all paths between sinks and sources that include a vertex or line, and it is the recommended normalization. Other options include dividing the number of paths containing a vertex or line by the maximum found among the vertices or lines (option *Normalize–Max*), which is useful when all traversal weights according to flow are low, and taking the logarithm of the number of paths containing a vertex or line (option *Logarithmic Weights)*, which is useful when the variation among traversal weights is very high. Finally, it is possible not to normalize the raw counts (option *Without Normalization)*. Note, however, that normalization does not affect the main paths that are later retrieved from the citation network with traversal weights computed. It merely changes the range and variation among traversal weights.

*Network> Info> Line Values*

The traversal weights of the papers (the original vertices) are stored in a vector, and the weights of the citations (lines) are saved as line values in a new network (labeled "Citation weights"), which can be inspected with the *Network> Info> Line Values* command.

When we apply the *Search Path Count (SPC)* command to the centrality literature network, about 90 percent of the lines have a traversal weight of 0.05 or less and thirteen lines have a value exceeding 0.103 (Table 22:

be sure the network labeled "Citation weights (SPC)" is selected in the drop-down menu when you execute the *Network> Info> Line Values* command and request #9 clusters). Clearly, one citation is very important to the development of the centrality literature: It has an extremely high traversal weight of 0.41. This is the citation of Bavelas' 1948 article by Leavitt in 1951. Bavelas (1948) and Leavitt (1951), as well as Freeman (1979) and Flament (1963), are the vertices with the highest traversal weights. These are the crucial articles in the centrality literature.

*Vector> Info*

Sometimes we want to know which articles (not citations) are the most important in the citation network. As we have already mentioned traversal weights of articles are stored as a Vector. Be sure that Vector labelled "Citation weights (SPC)" is selected in the drop-down menu when you execute the command *Vector> Info*. Input 10 in the first dialog box to get 10 most important articles according to traversal weights. In the centrality literature network the result is not surprizing: the most important article is the article by Bavelas (1948) with traversal weight 0.71, following by Freeman (1979) with traversal weight (0.56), Flament (1963) with traversal weight 0.46, and Leavitt (1951) with traversal weight 0.41. All other articles have traversal weights lower than 0.40.

The *Traversal Weights* commands do not automatically identify the main paths in the citation network. After computing the traversal weights, we must apply commands available in *Network> Acyclic Network> Create (Sub)Network> Main Paths* to obtain the main paths. Note that a network with traversal weights must be selected before searching for main paths.

*Network> Acyclic Network> Create (Sub)Network> Main Paths> Local Search> Forward*

We find the forward local main path with the *Network> Acyclic Network> Create (Sub)Network> Main Paths> Local Search> Forward* command. A dialog box allows us to set the tolerance, which we usually leave at 0. The command creates a partition identifying the vertices on the main paths (cluster 1) in the original citation network, and it produces a new network labeled "Forward Local Main Path" that contains the main paths. In the centrality literature, the main paths start with Bavelas (1948), proceed to Leavitt (1951), and end with Freeman (1977 and 1979); see the top main path in Figure 106. As an exercise, find the local forward main path with nonzero tolerance. How does the extracted main path differ from the one in Figure 106 (top)?

*Network> Acyclic Network> Create (Sub)Network> Main Paths> Global Search> Standard*

With *Standard global main path search*, we get a slightly different result; see the main path in the middle of Figure 106. The main path still starts with Bavelas (1948), proceeds to Leavitt (1951), and HeisseM (1951), but then proceeds to ChrisLM (1952) instead of proceeding to Shaw…C (1954). If we follow the standard global main path further, we get four more articles that are not present in the forward local main path (MacyCL (1953) – LuceMCH (1953) – ChristiB(1954) – LanzetR

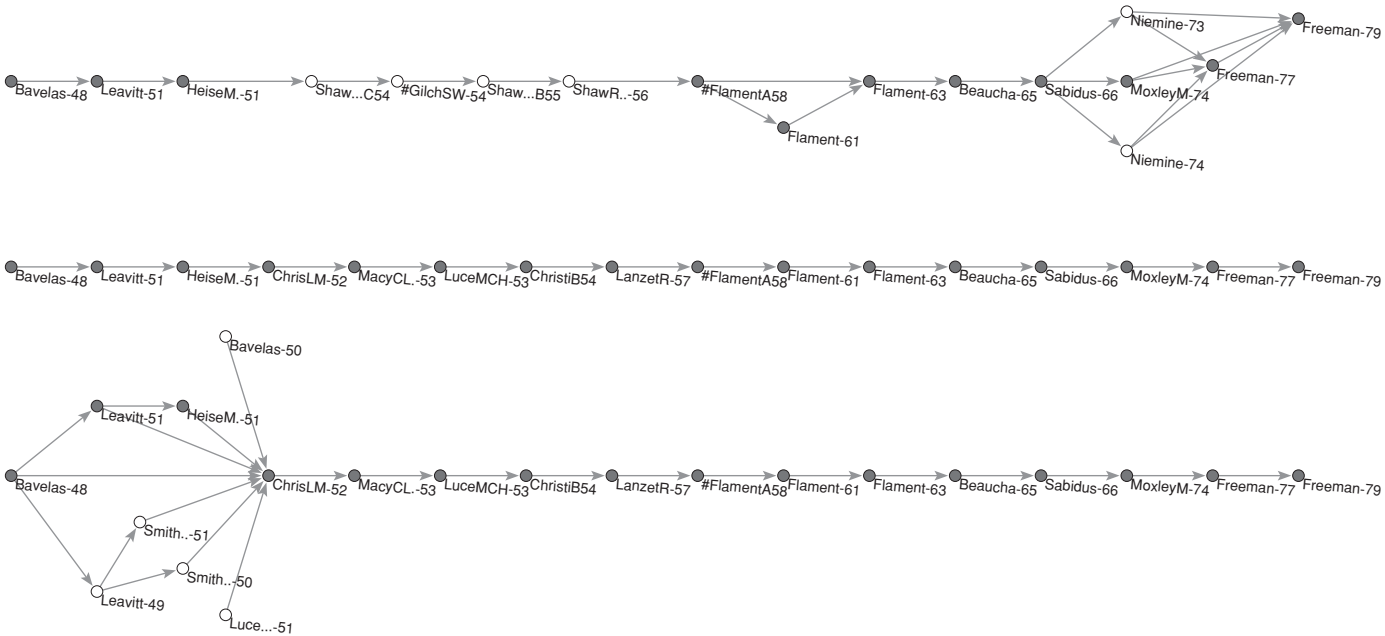



Figure 106. Forward local and key-route local (top), standard global and key-route global (middle), and backward local (bottom) main paths in the centrality literature network.

(1957)), but after that we reach the article FlamentA (1958), which is present also in the forward local main path. The rest of the standard global main path is the same as one single path that is part of the forward local main path, namely: Flament (1961) – Flament (1963) – Beaucha (1965) – Sabidusi (1966) – MaxleyM (1974) – Freeman (1977) – Freeman (1979).

*Network> Acyclic Network> Create (Sub)Network> Main Paths> Local Search> Key-Route*

Note that both main paths obtained so far contain the citation with the highest traversal weight (Bavelas' 1948 article cited by Leavitt in 1951). The Bavelas 1948 article is also the source paper – it does not cite any other paper. As a consequence, key-route local main path search with this citation as the only key-route yields the same result as forward local main path search (the top main path in Figure 106). We select the Bavelas 1948 paper citation by entering 1 behind *Select rank numbers of key-routes* in the dialog box of the *Network> Acyclic Network> Create (Sub)Network> Main Paths> Local Search> Key-Route* command. This tells Pajek to use the arc (citation) with the highest traversal weight as the only key-route or, more accurately, to use the arc at rank 1 in the list of line values. We can select the ten arcs with highest traversal weights by entering 1–10 in the dialog box.

*Network> Info> General*

*Network> Info> Line → Rank of its Value*

If several vertices have maximum traversal weight, Pajek selects the one reported first in *Network> Info> General* (when entering some positive number in the dialog box). To be sure that we have the citation of Bavelas 1948 by Leavitt in 1951, we should check the rank of this arc in the network with traversal weights. We can find the rank of a particular arc with the *Network> Info> Line → Rank of its Value* command. Enter the initial and terminal vertex number or label and the Report window displays the rank of this line according to its line value. In this way, we can find the rank of any special citation that we would like to use as a key-route. Enter the arc's rank number in the dialog box of the *Key-Route* command.

*Network> Acyclic Network> Create (Sub)Network> Main Paths> Global Search> Key-Route*

In this example, the key-route global main path search yields the same result as the standard global main path search if we select Leavitt's (1951) citation of the Bavelas 1948 article as the only key-route (the middle main path in Figure 106). Note that this need not be the case in other networks.

*Network> Acyclic Network> Create (Sub)Network> Main Paths> Local Search> Backward*

Finally let us apply the last search method: backward local main path search. The main path obtained by the backward local main path search contains the standard global main path as a subnetwork (see gray vertices in the bottom main path in Figure 106).

If we compare the results of forward local and backward local main path searches, we notice that forward local search adds more articles and citations at the end of the path – when we approach the Freemans articles – while backward local search adds some articles and

citations at the beginning of the path, soon after the Bavelas (1948) paper. If we would use some nonzero tolerance in searching for local main paths even more articles would be added to the local main paths.

If we need to select only one solution, we would probably select the one that appears most often. The main path obtained by the standard global main path search is obtained also using key route global main path search, and it is a subnetwork of the backward local main path search (gray vertices in the bottom main path). Several articles and citations in the forward local main path (and key-route local main path) are also present in the standard global main path (gray vertices in the top main path). Therefore the standard global main path seems to be the most appropriate for this example. But this needs not to be a general rule. For larger citations networks, we suggest to use nonzero tolerance and more key-routes to get more main paths, which may represent different research traditions.

Instead of searching for main paths containing selected key-routes (these are arcs representing citations) we can also search for main paths containing selected articles (represented by vertices). This can be done easily: first select one or more interesting articles (vertices numbers) in a Cluster and run the last command in this menu called: *Through Vertices in Cluster*. Again we can search for local and global main paths. For an example let us find main path containing both Leavitt (1949) and #GilchSW (1954) articles. As we can see in Figure 106 these two articles do not belong together to any main path. First create an empty Cluster (command *Cluster> Create Empty Cluster)* and manually edit the cluster so it contains the vertices 2 and 21 (2 and 21 are vertices numbers of Leavitt and #GilchSW articles respectively). Finally run command *Through Vertices in Cluster* (*Local* or *Global*). In case of global search the obtained main path contains 22 articles and in case of local search (zero tolerance) it contains 32 articles. Both main paths start in Bavelas (1948) article and finish in Freeman (1979) article what is again no surprize.

*Network> Acyclic Network> Create (Sub)Network> Main Paths> [Global, Local] Search> Through Vertices in Cluster*

*Cluster> Create Empty Cluster*

If *Mark Main Paths as Multirelational Network* is checked while searching main paths, the original network containing traversal weights is transformed into a multirelational network, where arcs belonging to the main path get relation number 2, while relation numbers for all other arcs are set to 1. A message in a Report window explains this. Be careful with this option: If the original network already contains multiple relations, these relations are overwritten.

*Network> Acyclic Network> Create (Sub)Network> Main Paths> Mark Main Paths as Multirelational Network*

The lowest traversal weight of the arcs in the main path is 0.05, but it is interesting to use a slightly lower cutoff value to obtain the main path component here. Let us delete all arcs with traversal weights lower

*Network> Create New Network> Transform> Remove> Lines with Value> lower than*

*Network> Create Partition> Components> Weak*

than 0.03. This can be done with the *Remove> Lines with Value> lower than* command in the *Network> Create New Network> Transform* submenu. Now, determine the weak components of minimum size 2 with the *Network> Create Partition> Components> Weak* command. The network contains two weak components, one large component with forty-six articles, a small component with three articles by Lawson and Burgess, and eighty isolated vertices.

*Operations> Network + Partition> Extract> SubNetwork Induced by Union of Selected Clusters*

*Partitions> Extract SubPartition (Second from First)*

Let us concentrate on the largest component and extract it with the *Operations> Network + Partition> Extract> SubNetwork Induced by Union of Selected Clusters* command, using the citation network with lines of minimum value 0.03 and the partition according to weak components. If we also extract the publication years of the forty-six articles in this component from the publication years partition (`centrality_literature_year.clu`) – select this partition as the first partition and the weak components partition as the second partition, and extract the first weak component (cluster 1) with the *Partitions> Extract SubPartition (Second from First)* command – we can draw this component into layers.

*Layers> Averaging x Coordinate*

*FishEye> Cartesian, Polar*

The resulting sociogram may look like Figure 107 if we optimize it with the *Layers> Averaging x Coordinate* command (*Forward* or *Backward)*. This figure reveals that the literature on network centrality was split into two lines between 1957 and 1979. One line was dominated by Cohen and the other by Flament and Nieminen. In 1979, Freeman integrated both lines in his classic article. If labels of vertices cannot be read because they overlap, put the Draw window into a FishEye mode (*FishEye> Cartesian, Polar*) as we have learned in Chapter 2.

*Operations> Network + Partition> Transform> Direction> Lower → Higher*

Mutual references among articles appearing at approximately the same time (e.g., two 1954 articles by Gilch in the original centrality network) or erroneous references to later articles by mistakes during data collection and coding may prevent the citation network from being acyclic. Then, the *Traversal Weights* commands issue a warning and stop; the network must first be made acyclic. References to later publications can be removed with the *Operations> Network + Partition> Transform> Direction> Lower → Higher* command (do not delete lines within clusters) provided that the partition according to publication dates was selected in the *Partition* drop-down menu.

*Network> Create Partition> Components> Strong*

*Operations> Network + Partition> Shrink Network*

In the centrality literature network, however, this solution did not work because both articles by Gilch appeared in 1954. In this case, we had to merge the articles. We computed the strong components of minimum size 2 (*Network> Create Partition> Components> Strong)* because they contain cyclically connected vertices in a directed network (see Chapter 10). We shrank each strong component to one vertex in a new network with the *Operations> Network + Partition> Shrink Network* command

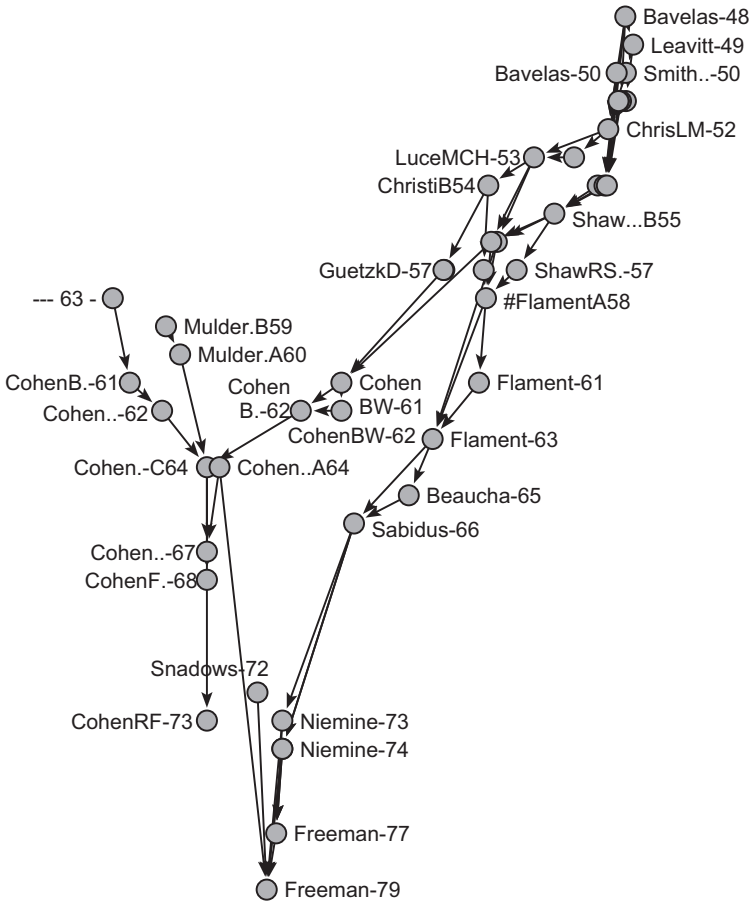



Figure 107. Main path component of the centrality literature network (not all names are shown here).

selecting 0 as the class that should not be shrunk because that class contained the vertices outside the strong components. We removed the loops with the *Network> Create New Network> Transform> Remove> Loops* command to obtain an acyclic network that allows the computation of citation weights.

*Network> Create New Network> Transform> Remove> Loops*

In the latest Pajek versions, however, it is much easier to transform a nearly acyclic network into an acyclic one: Just apply the *Preprint Transformation* command, which is available in menu *Network> Acyclic Network> Transform*. This command solves the problem of cyclic references by introducing so called "preprint vertices." Each vertex (paper)

*Network> Acyclic Network> Transform> Preprint Transformation*

inside a strong component is duplicated into a "preprint" version. Vertices (papers) inside each strong component cite "preprints."

## 11.7 Summary

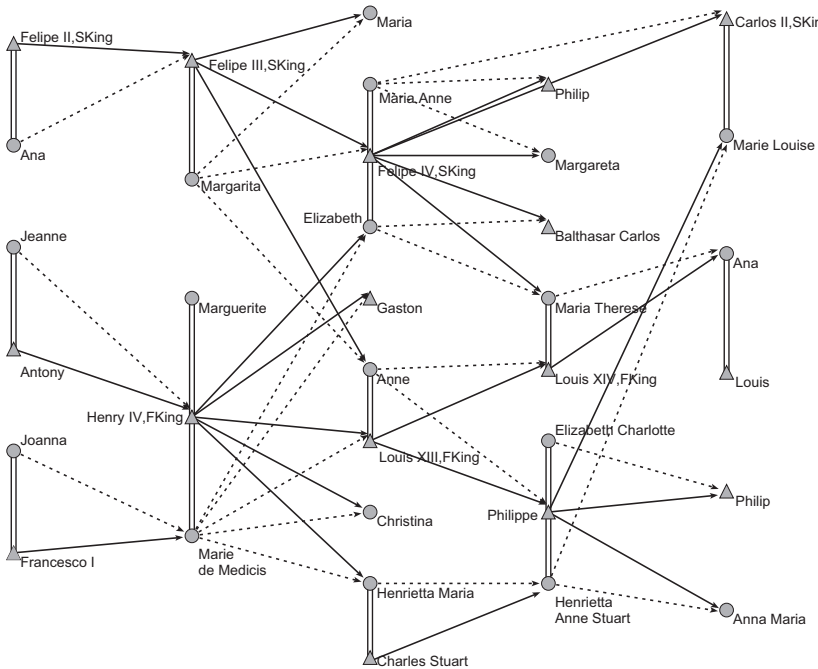
This was the last chapter that presented methods that cope with the dynamics of time in network analysis. Over time, social relations branch off into a gamut of independent strands. Kinship relations, for instance, create family trees that expand rapidly over generations. Sometimes, however, these strands merge after some time, for instance, people with common ancestors marry. This is called structural relinking, which is a measure of social cohesion over time. A social system with much relinking is relatively cohesive because relinking shows that people are oriented toward members of their own group or family.

In a genealogy, the amount of structural relinking can be assessed provided that we use a special kind of network: the P-graph. In contrast to an Ore graph, which represents each person by a vertex, parenthood by arcs, and marriage by (double) lines, couples and bachelors are vertices and individuals are arcs in a P-graph. Because symmetric marriages and parallel mother–child and father–child arcs are not represented by lines in the P-graph, each bi-component is an instance of structural relinking.

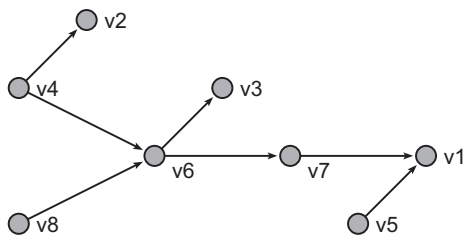
Methods for analyzing citation networks handle the time factor in a slightly different way. Here, we want to identify the publications that are the crucial links in the literature on a particular topic. Scientific articles contain knowledge, and citations indicate how knowledge flows through a scientific community. Each flow follows a path of citations, and citations that occur in many paths are important to the transmission of knowledge: They have high traversal weights. Citations with high traversal weights are linked into main paths, which represent the main lines of development in a research area. The articles and authors connected by citations of some minimum traversal weight constitute main path components, which are hypothesized to identify scientific specialties or subspecialties.

## 11.8 Questions

1. The Ore graph depicted below shows part of the family ties of Louis XIII, king of France (1601–43). Calculate the remove of his relation with Henrietta Anne Stuart.




2. Which people constitute the family of orientation of Louis XIII, and what is his family of procreation?
3. What is a generation jump? Indicate one in the Ore graph of Question 1.
4. Draw a P-graph that contains the same information as the Ore graph of Question 1.
5. How can we distinguish between a blood marriage and a relinking nonblood marriage in a P-graph? Give an example of both types of relinking in the genealogy of Louis XIII.
6. Explain why the relinking index of a tree is 0.
7. List all paths from sources to sinks in Figure 105, and show that the citation weight of the arc from v2 to v4 is correct.
8. Identify the source and sink vertices, the paths between them, and the traversal weight of the arcs in the following citation network below. What is the main path?



## 11.9 Assignment 1

The GEDCOM file `Isle_of_Man.ged` contains the combined genealogies of approximately twenty families from the British Isle of Man. Describe the overall structure of this network and the sections with structural relinking. Which types of relinking do occur?

## 11.10 Assignment 2

Publications and citations pass on scientific knowledge and traditions; so do advisors to their students. The file `PhD.net` contains the ties between Ph.D. students and their advisors in theoretical computer science; each arc points from an advisor to a student. The partition `PhD_year.clu` contains the (estimated) year in which the Ph.D. was obtained. Search for separate research traditions in this network and describe how they evolve.

## 11.11 Further Reading

- The genealogical data of the Ragusan nobility example were coded from the Ph.D. thesis of Irmgard Mahnken (1960): *Das Ragusanische Patriziat des XIV. Jahrhunderts*. For an analysis of a part of the genealogy, see V. Batagelj, "Ragusan families marriage networks." In A. Ferligoj and A. Kramberger (eds.), *Developments in Data Analysis* (Ljubljana: FDV, 1996, pp. 217–28).
- For the collection and storage of genealogical data, we advise to use the GEDCOM 5.5 standard (http://homepages.rootsweb.ancestry.com/~pmcbride/gedcom/55gctoc.htm). Good free software is Brothers Keeper available at http://www.bkwin.org, and Personal Ancestral File, which is produced and distributed by the Church of Jesus Christ of Latter-Day Saints (www.familysearch.org). This organization compiles a large database of genealogical information from which downloads can be made.
- The P-graph was presented by D. R. White and P. Jorion in "Representing and analyzing kinship: A network approach." *Current Anthropology* 33 (1992), 454–62; and in "Kinship networks and discrete structure theory: Applications and implications." *Social Networks* 18 (1996), 267–314.
- For additional reading on the analysis of kinship relations in the social sciences, we refer to T. Schweizer and D. R. White, *Kinship, Networks, and Exchange* (Cambridge: Cambridge University Press, 1998).

- The centrality literature example was taken from N. P. Hummon, P. Doreian, and L. C. Freeman, "Analyzing the structure of the centrality-productivity literature created between 1948 and 1979." *Knowledge-Creation Diffusion Utilization* 11 (1990), 459–80. The different types of main path analysis stem from N. P. Hummon and P. Doreian, "Connectivity in a citation network: The development of DNA theory." *Social Networks* 11(1989), 39–63. E. Garfield's *Citation Indexing: Its Theory and Application in Science, Technology, and Humanities* (New York, NY: John Wiley & Sons, 1979) is a classic text on citation analysis.
- Details on Key-Route main path searches are explained in J. S. Liu and L. Y. Y. Lu, "An integrated approach for main path analysis: Development of the Hirsch index as an example." *Journal of the American Society for Information Science and Technology*, 63 (2012), 528–42. See also Wikipedia: https://en.wikipedia.org/wiki/Main_path_analysis
- The analysis of networks that can be derived from bibliographic data are discussed in V. Batagelj and M. Cerinšek, "On bibliographic networks." *Scientometrics*, 96 (2013) 3, 845–64 and in Chapters 3 and 4 of V. Batagelj, P. Doreian, A. Ferligoj, and N. Kejžar, *Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution* (New York, NY: John Wiley & Sons, 2014). Software for transforming bibliographic data into Pajek networks is available at https://github.com/bavla/biblio.

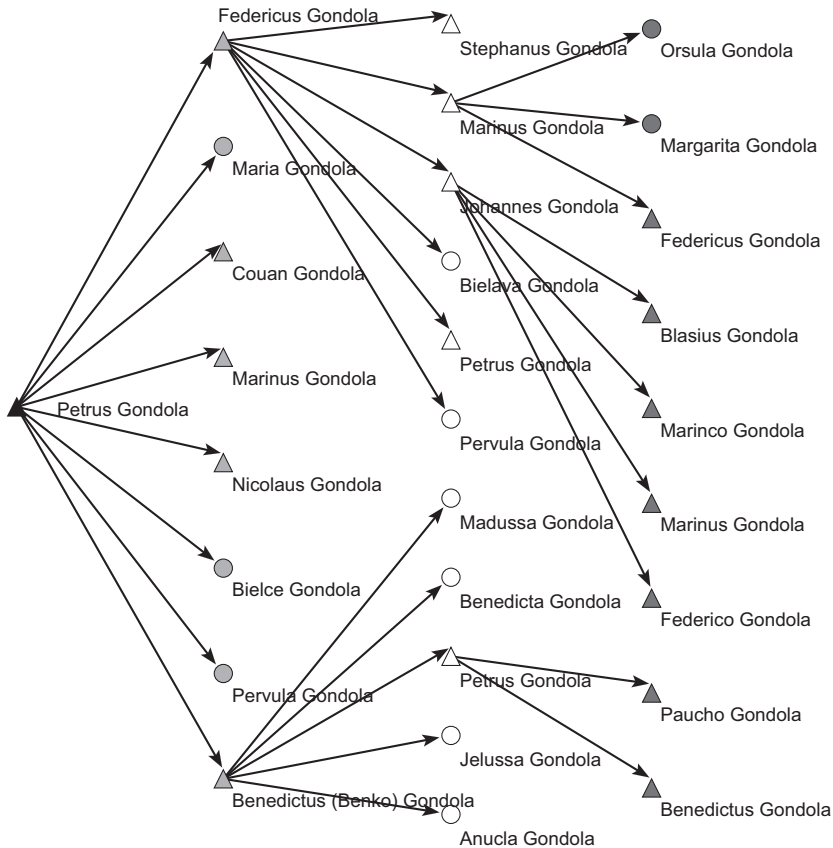
## 11.12 Answers

*Answers to the Exercises*

I. You should realize that a surname was passed on from father to child in Ragusa. Mother–child ties and marriages do not matter (we are concerned only with the name given at birth). Therefore, you should eliminate all marriages and mother–child ties from the `Gondola_Petrus.ged` data. This can be done easily if you open the GEDCOM file with the option *Options> Read – Write> Ore: Different relations for male and female links* selected. In this network, you must select the father–child relation (*Network> Multiple Relations Network> Extract Relation(s)* into Separate Networks, enter relation number 1).

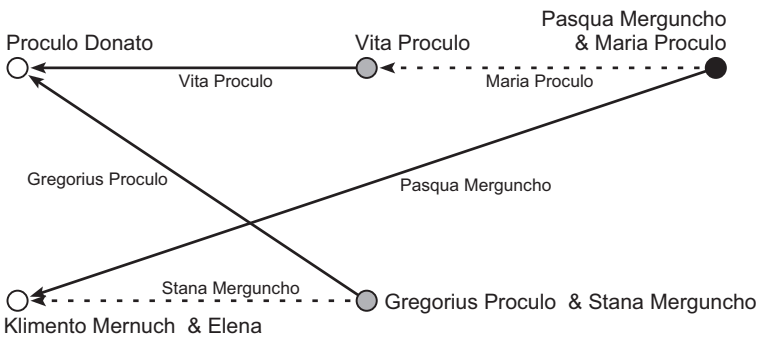
In the resulting network, the descendants of Petrus Gondola are all people who received his surname. Identify them with the *k-Neighbours> Output* command (Petrus Gondola has a vertex number of 94) and extract them from the network with the *Operations> Network + Partition> Extract> SubNetwork Induced by Union of*

*Selected Clusters* command (in the dialog box enter 0–3), making sure that the *k-Neighbours* partition is selected in the *Partition* drop-down menu. If you determine the genealogical depth partition of the new network (*Network> Acyclic Network> Create Partition> Depth Partition> Genealogical*), draw it in layers (*Layers> In y Direction*), optimize it (*Layers> Optimize Layers in x Direction> Forward*), and rotate it 90 degrees (*[Draw] Options> Transform> Rotate 2D)*, it should look like the sociogram that follows.



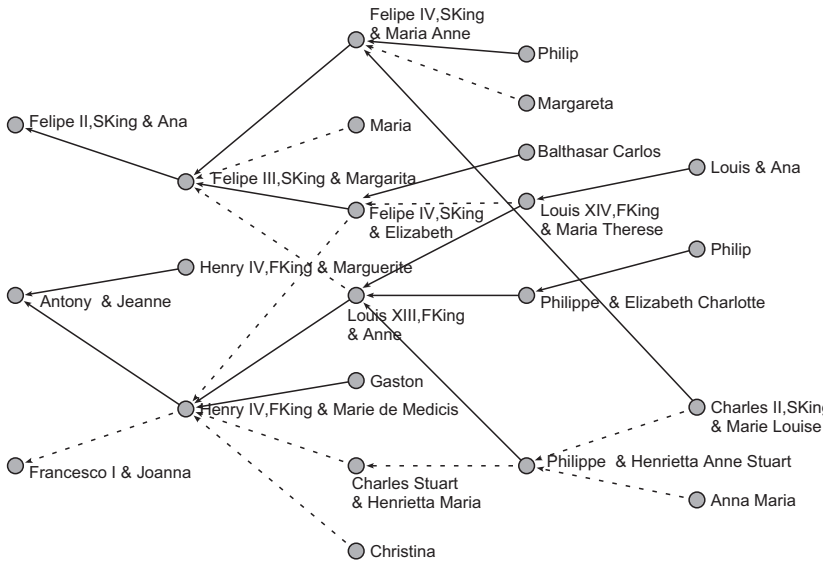

II. In the "Application" part of Section 11.4, you learned how to determine the bi-components in a P-graph (command: *Network> Create New Network> with Bi-Connected Components stored as Relation Numbers)*, how to create a cluster from a bi-component in the hierarchy of bi-components (*Hierarchy> Extract Cluster)*, and how to extract this component from the original P-graph (*Operations> Network + Cluster> Extract SubNetwork)*. In this way, you can obtain

the subnetwork of the small bi-component in the Ragusan nobility genealogy consisting of five vertices. With the vertices relocated, this network may look like the sociogram that follows. From this layout, it is clear that the marriages closing the semicycle are not blood marriages. Neither Pasqua Merguncho nor Maria Proculo has a close common ancestor, nor have Gregorius Proculo and Stana Merguncho. This structural relinking is an instance of repeated marriages between two families: The Proculo and Mernuch/Merguncho families swap a son.




*Answers to the Questions in Section 11.8*

1. Louis XIII is the uncle (mother's brother) of Henrietta Anne Stuart, so she is a relative in the third degree if we restrict ourselves to blood relations. Louis XIII is also her stepfather, so the degree is 1 if we include marital ties.
2. The family of orientation of Louis XIII includes his parents Henry IV and Marie de Medicis, his brother Gaston, and his sisters Elizabeth, Christina, and Henrietta Maria. Marguerite, the other wife of Henry IV, may or may not belong to the family of orientation. His family of procreation contains his wife Anne and their children Louis XIV and Philippe.
3. A generation jump in a genealogy refers to a relinking marriage that connects people of different genealogical generations, which are calculated from the point of view of their common ancestor. The marriage between Carlos II and Marie Louise creates a generation jump, because Carlos is a grandson of Felipe III and Margarita (second remove) and Marie Louise is the granddaughter of the daughter (Anne) of Felipe III and Margarita (third remove).
4. The P-graph should look like the following figure. Do not forget to draw different arcs for men and women and to reverse the direction of arcs.




5. In a P-graph, the husband and wife involved in a blood marriage share at least one ancestor: There are two paths from the blood marriage to an ancestor, for instance, from Philippe and Henrietta Anne Stuart to Henry IV, king of France, and his spouse Marie de Medicis. Both Philippe and Henrietta Anne Stuart are their grandchildren. A relinking nonblood marriage is a marriage between descendents of families that are already linked by intermarriage; for example, the Spanish king Felipe III and the French king Henry IV are linked by two marriages among their children: Felipe IV and Elizabeth, Louis, XIII, and Anne. In a P-graph, this type of relinking is characterized by two semipaths (or one path and one semipath) between couples.
6. Structural relinking involves semicycles: Vertices are connected by two paths or semipaths. Because trees contain no semicycles by definition, there is no relinking and the relinking index is 0.
7. The eight paths are as follows: (1) v1 → v3, (2) v1 → v4, (3) v1 → v2 → v4, (4) v1 → v6 → v4, (5) v1 → v6 → v2 → v4, (6) v5 → v6 → v4, (7) v5 → v6 → v2 → v4, and (8) v5 → v2 → v4. Four paths include the arc v2 → v4 (viz., paths 3, 5, 7, and 8), which is half of all paths, so the traversal weight of this arc is 0.5.
8. The source vertices are v4, v8, and v5; v2, v3, and v1 are sink vertices. There are six paths from sources to sinks as follows: (1) v4 → v2, (2) v4 → v6 → v3, (3) v4 → v6 → v7 → v1, (4) v8 → v6 → v3, (5) v8 → v6 → v7 → v1, and (6) v5 → v1. The arcs v4 → v2 and v5 → v1 are

included in one of these paths, so their traversal weight is 1 divided by 6 as follows: 0.167. The other arcs are included in two paths, so their traversal weights are 0.333. There are four main paths: (1) from v4 to v3, (2) from v4 to v1, (3) from v8 to v3, and (4) from v8 to v1.