Network Analysis for Humanities and Social Sciences

· a visual tutorial of complex systems ·

Antonio Rivero Ostoic

Aarhus University

University of Southern Denmark lecture ☆ 9 November 2021

Agenda

algebraic analysis & visualization . . .

1. Plotting multigraphs (showcase)
   - ➠ Example 1: Monastery novices
2. Group structure in social networks
   - ➠ Example 2: Kariera society kinship network
3. Complex organisational networks
   - ➠ Example 3: Multilevel structure of the Group of Twenty
4. Network analysis in history and archaeology
   - ➠ Work-in-progress: Ancient Roman world

1. Plotting multigraphs (showcase)

Example 1: Monastery novices

Package 'multiplex'

August 28, 2013

Type Package

Title Analysis of Multiple Social Networks with Algebra

Version 1.0

Depends R (>= 3.0.1)

Date 2013-08-28

Author J. Antonio Rivero Ostoic

Maintainer Antonio Rivero Ostoic <multiplex@post.com>

Description multiplex - Analysis of Multiple Social Networks with Algebra is a package for the study of social systems made of different types of relationships. It is possible to create and manipulate multivariate network data with different formats, and there are effective ways available to treat multiple networks with routines that combine algebraic systems like the partially ordered semigroup or the semiring structure together with the relational bundles occurring in different types of multivariate network data sets.

License GPL-3

Suggests Rgraphviz

Encoding latin1

Collate
'as.semigroup.R' 'as.strings.R' 'bundle.census.R' 'bundles.R''cngr.R' 'convert.R' 'cph.R' '

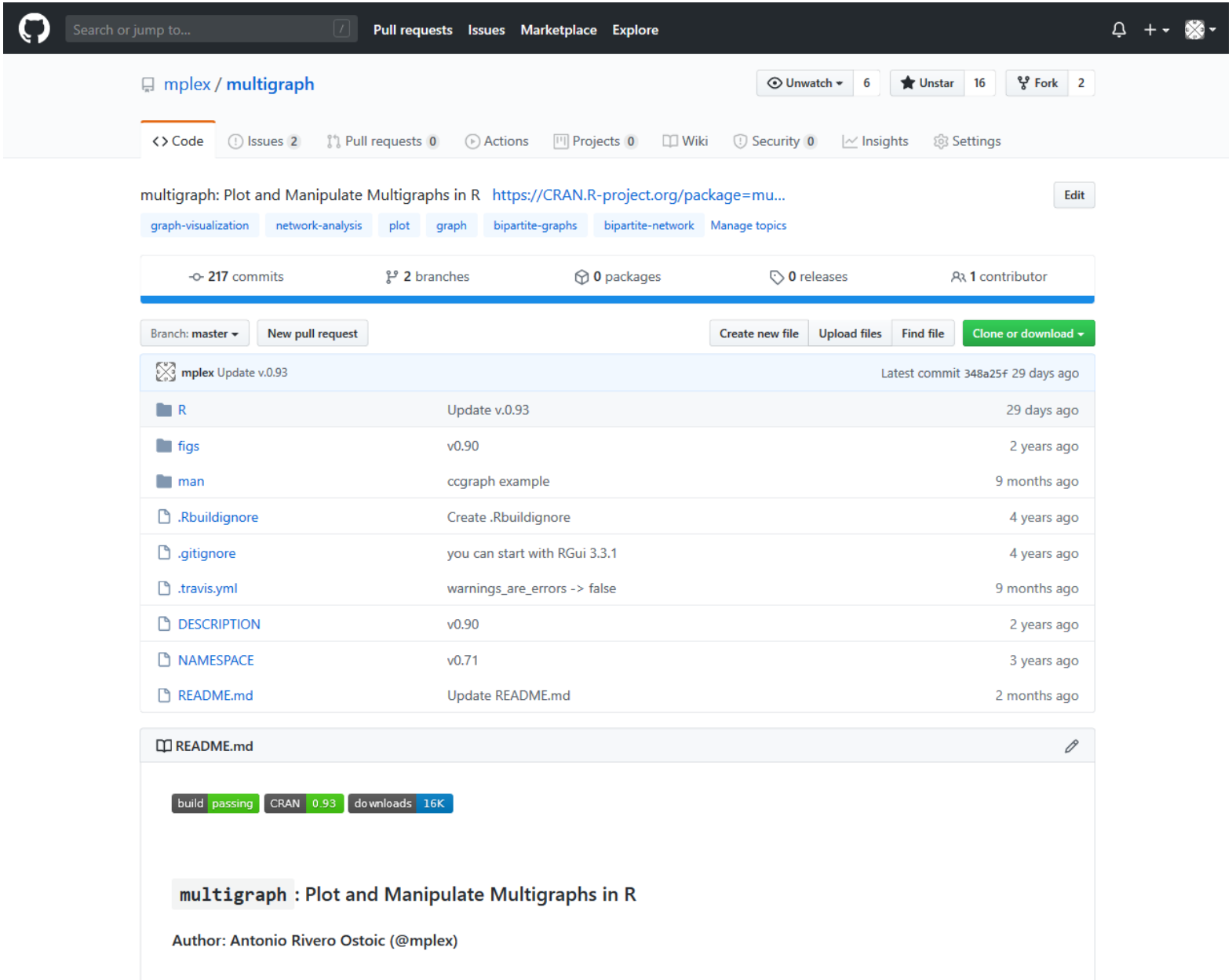
NeedsCompilation no

Repository CRAN

Date/Publication 2013-08-28 13:53:11

R topics documented:

‘multigraph’ *to depict multiplex networks in* ***R***



Getting started

Run `multiplex` and `multigraph` online:

https://rdrr.io/cran/multiplex/

https://rdrr.io/cran/multigraph/

Or download **R** and install packages:

```
# install 'multiplex' & 'multigraph' from CRAN
install.packages("multiplex")
install.packages("multigraph")
```

```
# load packages
library("multigraph")
# Loading required package: multiplex
```

Representing network data

There are different ways to represent network data in **R**

$(1, 2)$

```
transf("1, 2")
```

```
  1 2
1 0 1
2 0 0
```

```
multigraph("1, 2", cex = 18, lwd = 20, rot = -90, pos = 0, vedist = -2)
```

```
scp <- list(cex = 18, lwd = 20, rot = -90, pos = 0, vedist = -2)
multigraph("1, 2", scope = scp)
```

Representing network data

Undirected

$\{1, 2\}$

```
matrix(c(0,1,1,0), nrow = 2, ncol = 2)
```

```
     [,1] [,2]
[1,]    0    1
[2,]    1    0
```

```
multigraph("1, 2", directed = FALSE, scope = scp)
```

Multiplex

$(1,2);(2,1)$

```
, , 1

  1 2
1 0 1
2 0 0

, , 2

  1 2
1 0 0
2 1 0
```

```
multigraph(list("1, 2", "2, 1"), scope = scp, ecol = 1, bwd = .7)
```

Multiplex

$(1,2);(2,1)$

```
, , 1

  1 2
1 0 1
2 0 0

, , 2

  1 2
1 0 0
2 1 0
```

```
net <- list("1, 2", "2, 1")
multigraph(net, scope = scp, ecol = 1, bwd = .7, swp = TRUE)
```

Monastery novices: Directed, multiplex, signed, valued, and longitudinal

```
# Read Sampson Monastery dataset as Ucinet DL file (on a mirror)
samp <- read.dl("http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/sampson.dat")
```

```
# What types of tie the network has?
dimnames(samp)[[3]]
```

```
[1] "SAMPLK1" "SAMPLK2" "SAMPLK3" "SAMPDLK" "SAMPES"  "SAMPDES" "SAMPIN"  "SAMPNIN" "SAMPPR"  "SAMNPR"
```

➠ "like T1-T3", "dislike", "esteem", "disesteem", "influence" (pos/neg), "praise" (pos/neg)

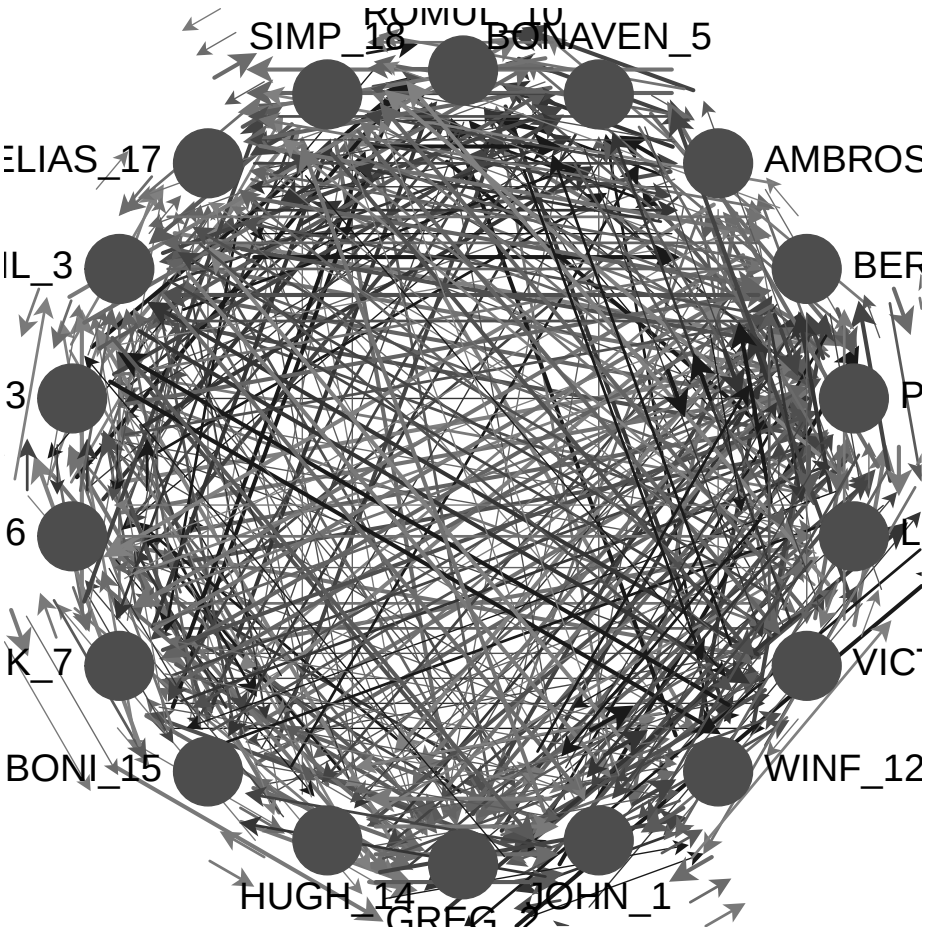
```
# Plot Monastery novices network as valued multigraph (default)
multigraph(samp, valued = TRUE)
```

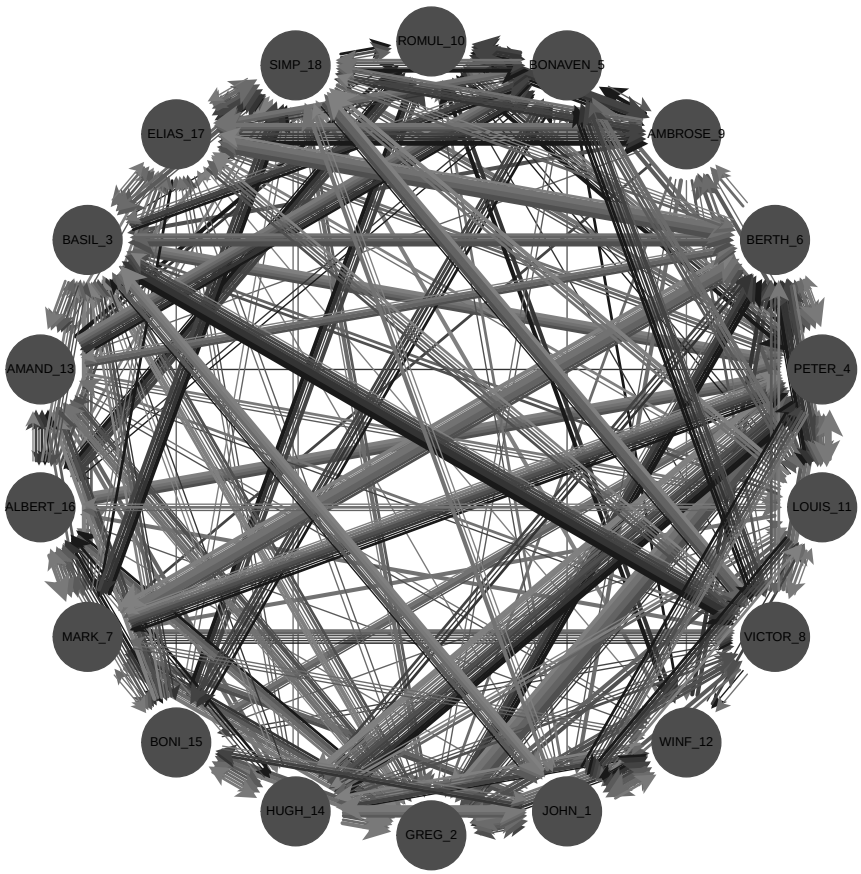
```
# plot valued network with customized values
multigraph(samp, valued = TRUE, bwd = .1, pos = 0, fsize = 6)
```

(Sampson, 1969)

Monastery novices network plot

multigraph circular





Monastery novices: Bundle patterns

```
# enumeration of bundle class types
bundle.census(samp)
```

```
      BUNDLES NULL ASYMM RECIP T.ENTR T.EXCH MIXED FULL
TOTAL     134   19    20     1     37      8    68    0
```

```
# bundle patterns in the Monastery novices network
summaryBundles(bundles(samp))
```

```
                                                                                  Bundles
Asym1                                                          ->{SAMPLK1} (WINF_12, BONAVEN_5)
Asym2                                                           ->{SAMPLK1} (BASIL_3, ROMUL_10)
...                                                                                       ...
Asym20                                                           ->{SAMNPR} (AMAND_13, SIMP_18)
Recp                                                           <->{SAMPLK3} (BONI_15, VICTOR_8)
Tent1                                 ->{SAMPDLK} ->{SAMPDES} ->{SAMNPR}  (ALBERT_16, ELIAS_17)
Tent2                      ->{SAMPDLK} ->{SAMPDES} ->{SAMPNIN} ->{SAMNPR}  (ALBERT_16, PETER_4)
...                                                                                       ...
```

Monastery novices: Define a system & depict

```
# recall network types of tie
dimnames(samp)[[3]]
```

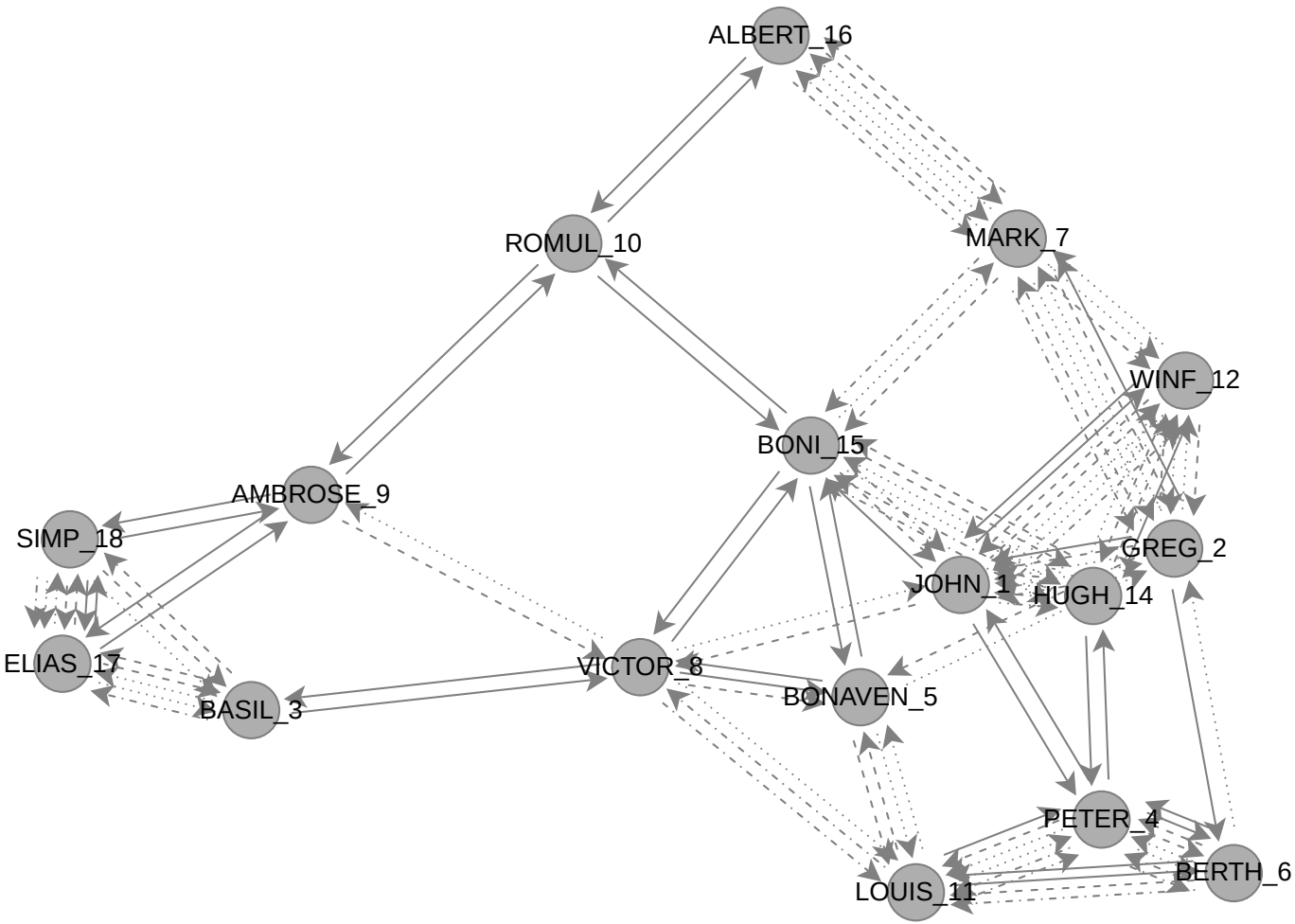
➠ "like T1-T3", "dislike", "esteem", "disesteem", "influence" (pos/neg), "praise" (pos/neg)

```
# Extract system of strong bonds having positive ties
sampsb <- rel.sys(samp[,,c(3,5,7,9)], type = "toarray", bonds = "strong")
```

```
# define a new scope
scps <- list(fsize = 8, pos = 0, vcol = "#AEAEAE", vcol0 = "#808080",
             ecol = "#808080", bwd = .4, rot = 145, mirrorY = TRUE)

# plot graph with a reproducible force-directed layout
multigraph(sampsb, layout = "force", seed = 12, scope = scps)
```

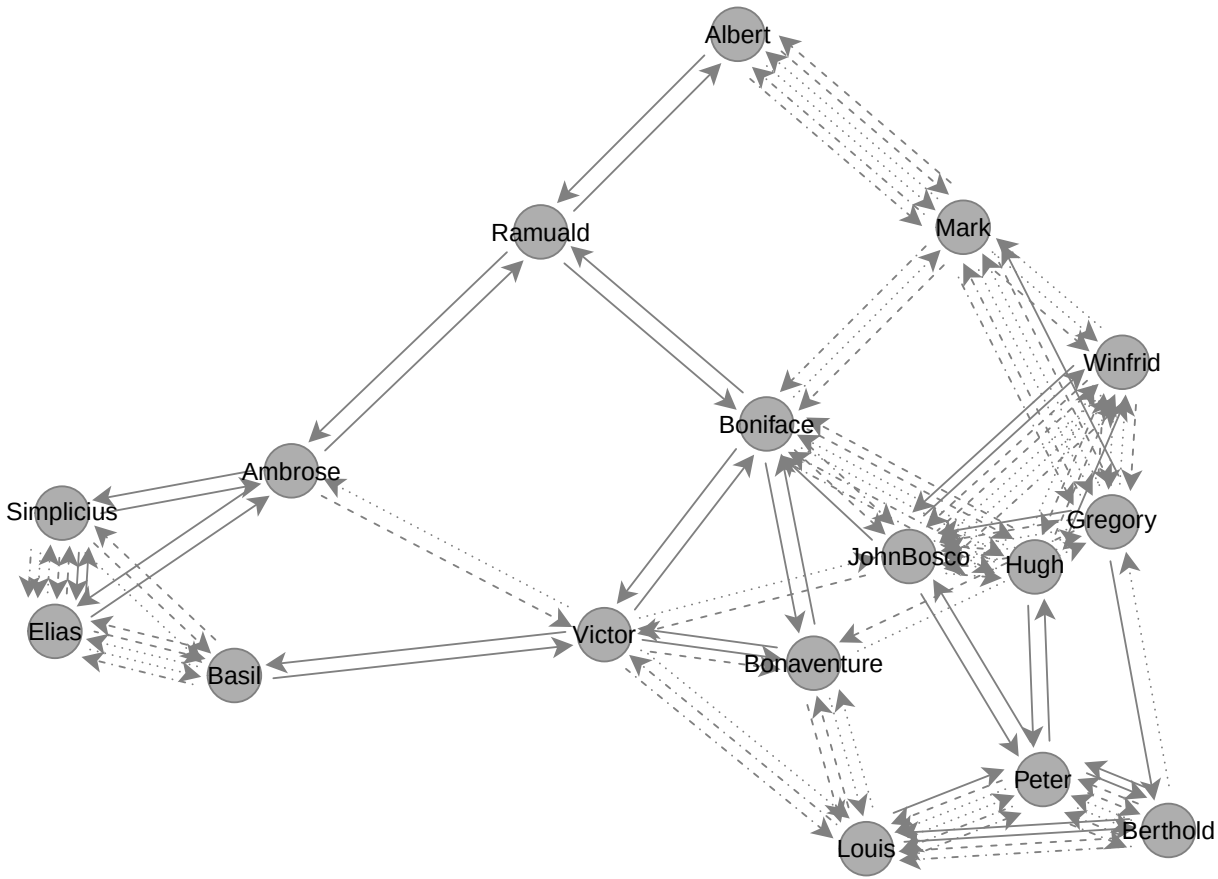
System of strong bonds

Monastery novices network



System of strong bonds: Customized node labels

```
monks <- c("Ramuald","Bonaventure","Ambrose","Berthold","Peter",
+           "Louis","Victor","Winfrid","JohnBosco","Gregory","Hugh",
+           "Boniface","Mark","Albert","Basil","Elias","Simplicius")

multigraph(sampsb, layout = "force", seed = 12, scope = scps, lbs = monks)
```



2. Group structure in social networks

Example 2: Kariera society kinship network

Kariera society kinship system and group structure

algebraic groups to model human societies

- Some primitive societies like the *Kariera* from Western Australia have kinship networks that follow the rules of a group structure

  ➠ where "primitive" means "first of its class"

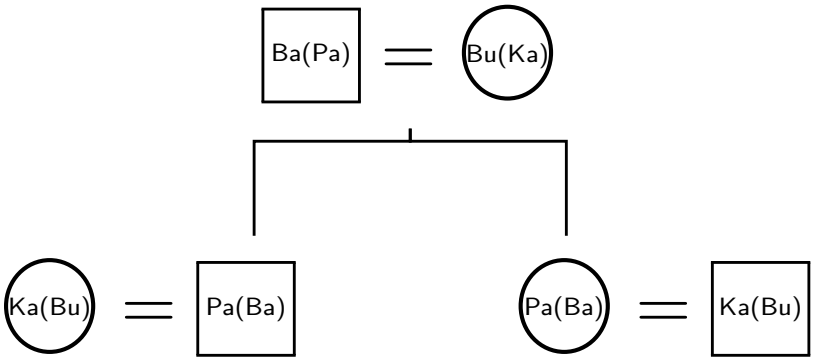
- The Karieras have (had?) four clans with specific rules of marriage & descent: *Banaka*, *Burung*, *Karimera*, and *Palyeri*.

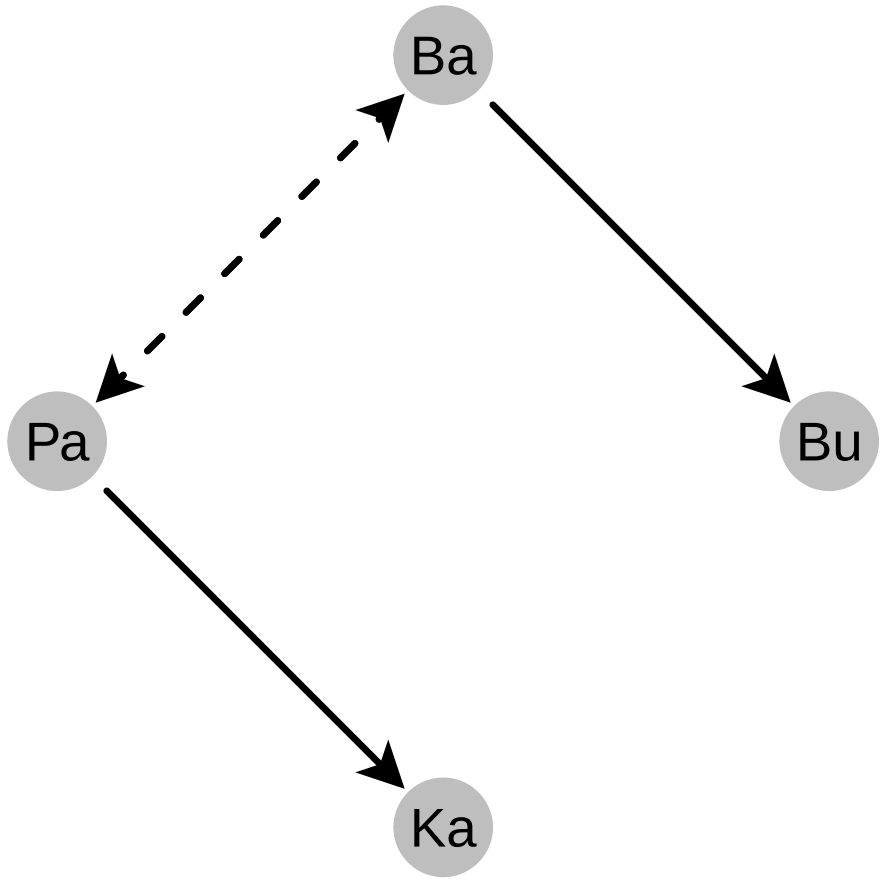
  ➠ Data collected by Radcliffe-Brown (1913), analysed in White (1963)

Kariera rules for marriage & descent (I)

Banaka (Ba), Burung (Bu), Karimera (Ka), Palyeri (Pa)

Two types of descent rules among Banaka and Palyeri (ego male)

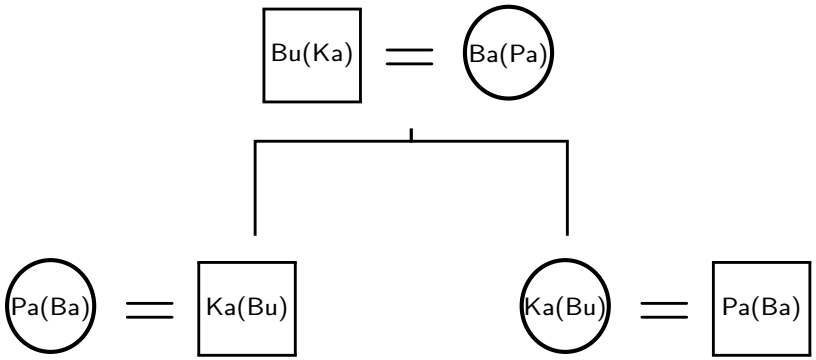


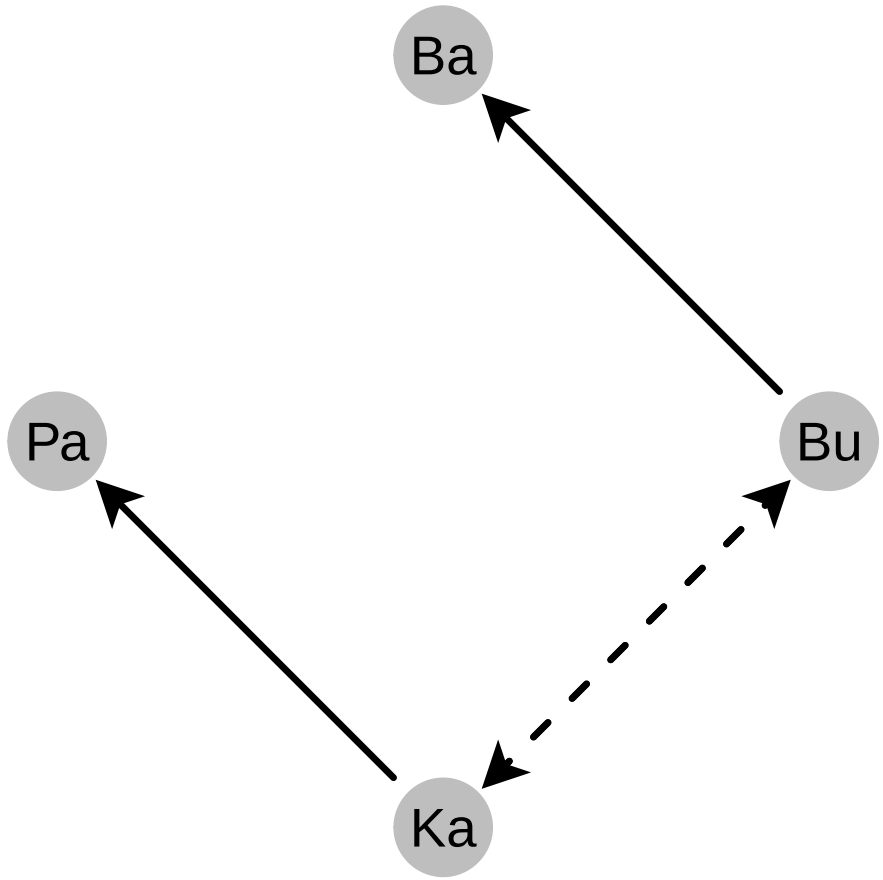


Kariera rules for marriage & descent (II)

Banaka (Ba), Burung (Bu), Karimera (Ka), Palyeri (Pa)

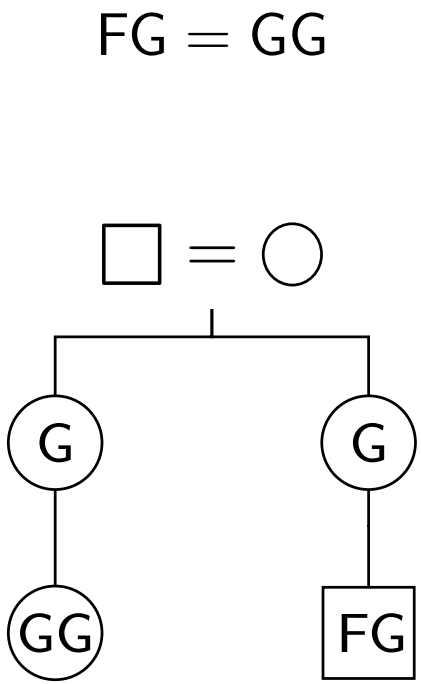
Two types of descent rules among Burung and Karimera (ego male)



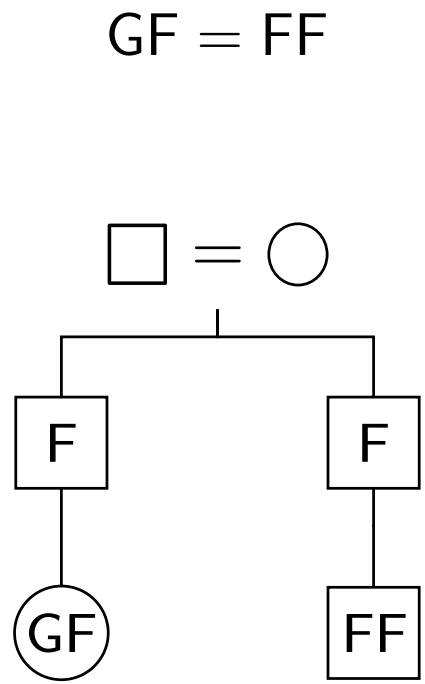


Parallel-cousins marriages in kinship networks

identifiers, F for male and G for female, are with right multiplication



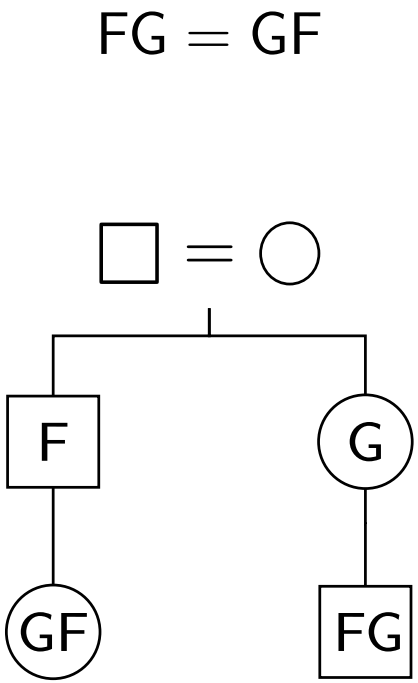
(a) Matrilineal



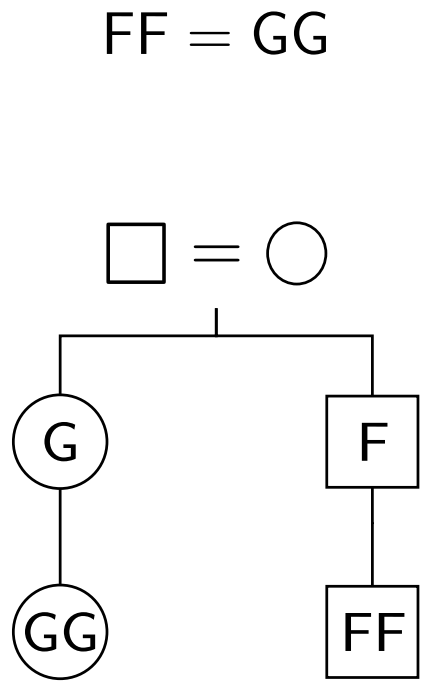
(b) Patrilineal

Cross-cousins marriages in kinship networks

identifiers, F for male and G for female, are with right multiplication



(a) Matrilineal



(b) Patrilineal

Permutation matrices for marriage & descent

Kariera kinship system

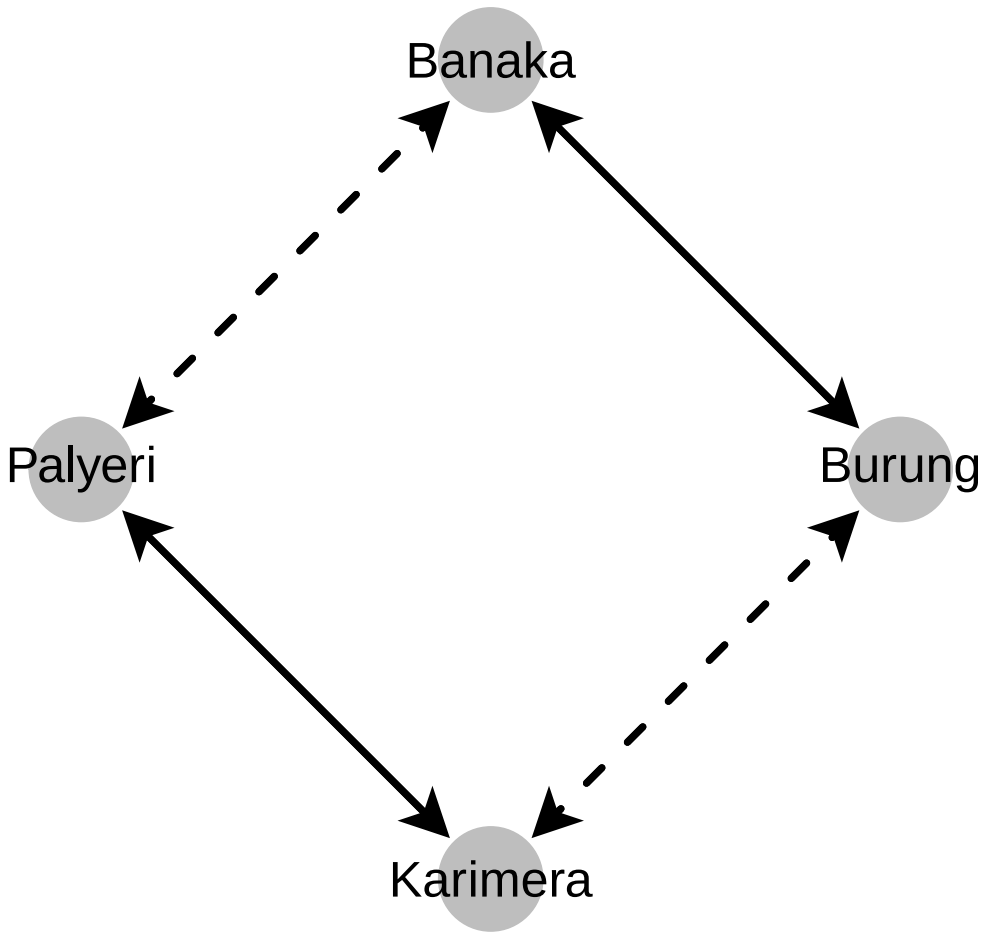
```
# create permutation matrices for marriage & descent rules
kks <- transf(list(F = c("1, 2", "3, 4", "2, 1", "4, 3"),
+     G = c("1, 4", "2, 3", "3, 2", "4, 1")))
```

```
, , F

  1 2 3 4
1 0 1 0 0
2 1 0 0 0
3 0 0 0 1
4 0 0 1 0

, , G

  1 2 3 4
1 0 0 0 1
2 0 0 1 0
3 0 1 0 0
4 1 0 0 0
```

Rules of marriage & descent

Kariera kinship system

```
# visualize marriage & descent rules in the Kariera
multigraph(kks, scope = scpD3, ecol = 1, collRecip = TRUE,
+    lbs = c("Banaka", "Burung", "Karimera", "Palyeri"))
```



Set of equations

to identify cross- and parallel-cousins marriages

The *set of equations* to detect allowed marriage types by commutation

```
# the equations allows finding marriage types in 'kks'
strings(kks, equat = TRUE)
```

```
...
$st
[1] "F"  "G"  "FF" "FG"

$equat
$equat$FF
[1] "FF" "GG"

$equat$FG
[1] "FG" "GF"


$equate
$equate$e
[1] "e"  "FF" "GG"
```

☛ *Both cross-cousins marriages are permitted in the Kariera*

Group structure as multiplication table

Kariera kinship system

The *multiplication table* reflects the group structure of the clan system

```
# Group structure with a symbolic format
semigroup(kks, type = "symbolic")
```

```
$dim
[1] 4
...

$ord
[1] 4

$st
[1] "F"  "G"  "FF" "FG"

$S
    F  G FF FG
F  FF FG  F  G
G  FG FF  G  F
FF  F  G FF FG
FG  G  F FG FF

attr(,"class")
[1] "Semigroup" "symbolic"
```

Algebraic constraints in group structures

Two *algebraic constraints* for the analysis of the elementary structures:

- *Set of equations* among different types of tie
- *Multiplication table* with relations between the different types of tie

☛ Complex structures have additional algebraic constraints

Hierarchy of relations: Partial order structure

Complex structures with lack of symmetry

With semigroups typically there exists an ordering among its relations

- A *partial order* is defined by an *inclusion* relation $\leq$ among $x, y \in S(R)$ with the rule:

$$S(R)^{\leq}_{x,y} = \begin{cases} 1 & \text{iff relation } x \text{ is contained in relation } y \\ 0 & \text{otherwise} \end{cases}$$

where 'contained' implies that all ties in x are occurring in y as well

➠ A *partially ordered semigroup* is $S(R)$ with a partial order

More on modeling complex networks with a partially ordered semigroup

Journal of Statistical Software

February 2020, Volume 92, Issue 11. *doi: 10.18637/jss.v092.i11*

Algebraic Analysis of Multiple Social Networks with multiplex

J. Antonio Rivero Ostoic
Aarhus University

Abstract

multiplex is a computer program that provides algebraic tools for the analysis of multiple network structures within the R environment. Apart from the possibility to create and manipulate multivariate data representing multiplex, signed, and two-mode networks, this package offers a collection of functions that deal with algebraic systems – such as the partially ordered semigroup, and balance or cluster semirings – their decomposition, and the enumeration of bundle patterns occurring at different levels of the network. Moreover, through Galois derivations between families of the pairs of subsets in different domains it is possible to analyze affiliation networks with an algebraic approach. Visualization of multigraphs, different forms of bipartite graphs, inclusion lattices, Cayley graphs is supported as well with related packages.

Keywords: social network analysis, relational algebra, graph visualization, R.

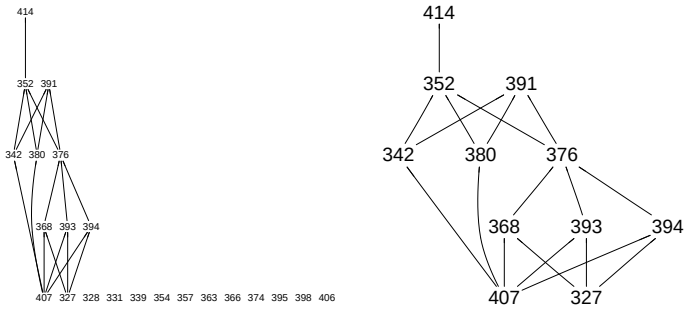


Figure 3: CPH of Incubator C with and without incomparable elements in the poset.

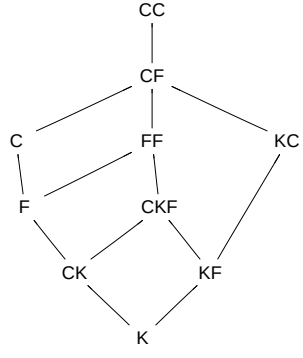


Figure 5: Hierarchy of string relations in the role structure of `netC`.

3. Complex organisational networks

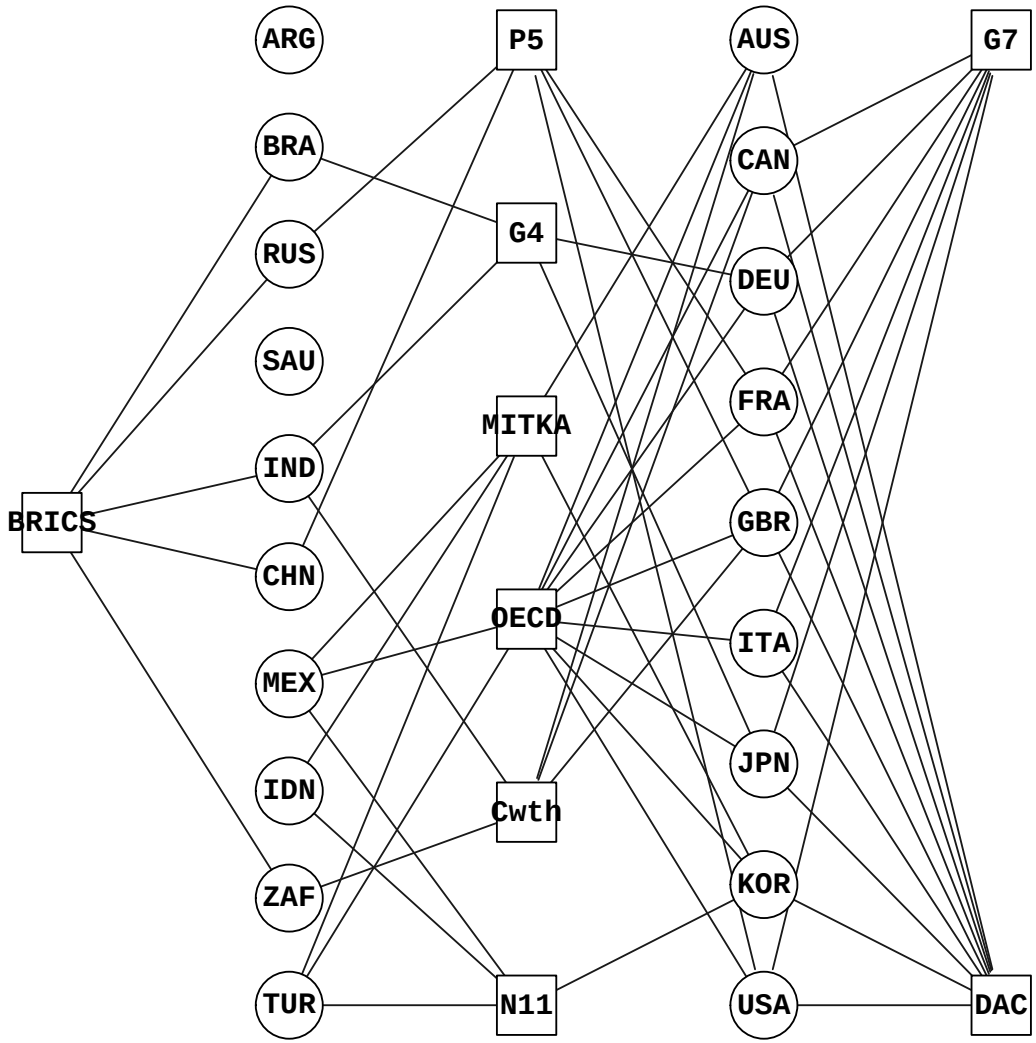
Example 3: Multilevel structure of the Group of Twenty

Group of Twenty

1. Argentina
2. Australia
3. Brazil
4. Canada
5. China
6. France
7. Germany
8. India
9. Indonesia
10. Italy
11. Japan
12. Mexico
13. Russia
14. Saudi Arabia
15. South Africa
16. South Korea
17. Turkey
18. United Kingdom
19. United States
20. (European Union)

Group of Twenty (G20) affiliations

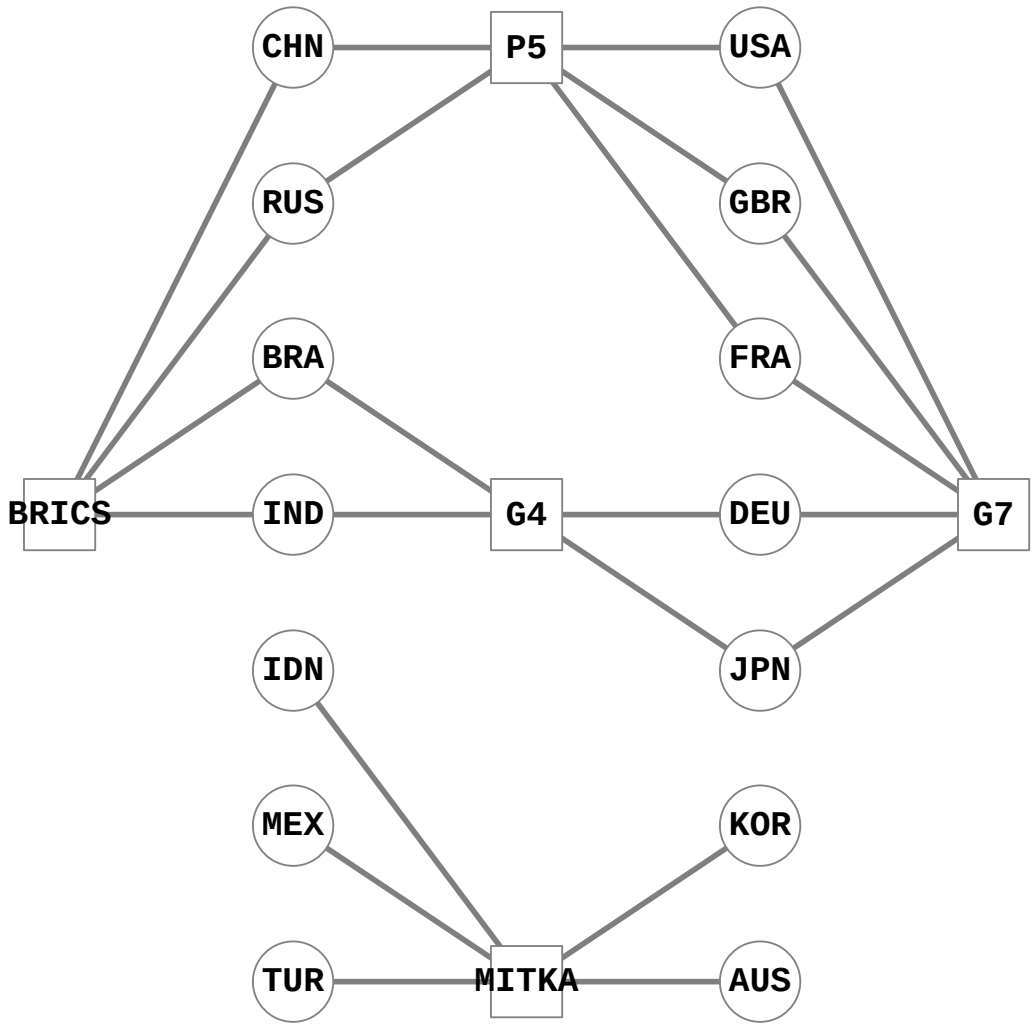
bipartite graph



(affiliations from Wikipedia)

G20 with non-overlapping "bridge" organisations

bipartite graph



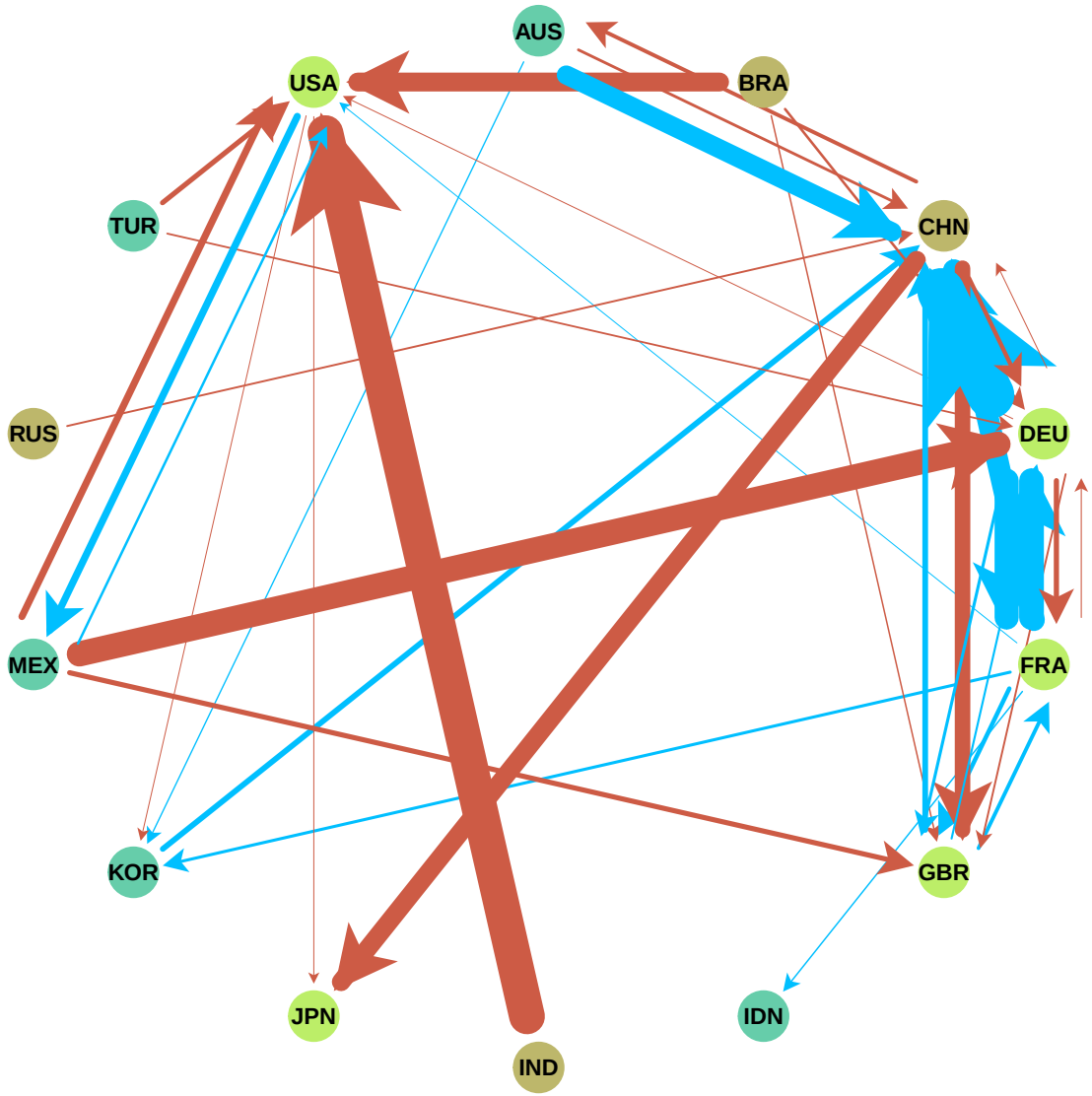
Classes of actors in G20 “bridges” (G20B)

1. G7
2. BRICS
3. MITKA

AUS	BRA	CHN	DEU	FRA	GBR	IDN	IND	JPN	KOR	MEX	RUS	TUR	USA
3	2	2	1	1	1	3	2	1	3	3	2	3	1

Skeleton trade network of milk & homey among G20B

multigraph circular layout



(based on UN COMTRADE, 2017)

Constructing G20 affiliation network

```r
# construct 'G20' dataset with events and actors
G20 <- data.frame(
        P5    = c(0,0,0,0,1,0,1,1,0,0,0,0,0,0,1,0,0,1,0),
        G4    = c(0,0,1,0,0,1,0,0,0,1,0,1,0,0,0,0,0,0,0),
        G7    = c(0,0,0,1,0,1,1,1,0,0,1,1,0,0,0,0,0,1,0),
        BRICS = c(0,0,1,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,1),
        MITKA = c(0,1,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0),
        DAC   = c(0,1,0,1,0,1,1,1,0,0,1,1,1,0,0,0,0,1,0),
        OECD  = c(0,1,0,1,0,1,1,1,0,0,1,1,1,1,0,0,1,1,0),
        Cwth  = c(0,1,0,1,0,0,0,1,0,1,0,0,0,0,0,0,0,0,1),
        N11   = c(0,0,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0) )
# actors names
rownames(G20) <- c("ARG","AUS","BRA","CAN","CHN","DEU","FRA","GBR","IDN","IND",
                   "ITA","JPN","KOR","MEX","RUS","SAU","TUR","USA","ZAF")
```

```r
# event clustering information
ec <- c(1, 1, 2, 0, 1, 2, 1, 1, 1)

# actor clustering (IMF economic classification of countries)
ac <- c(0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0)
ac <- replace(ac, ac==0, "Emerging")
ac <- replace(ac, ac==1, "Advanced")
```

```
 [1] "Emerging" "Advanced" "Emerging" "Advanced" "Emerging" "Advanced" "Advanced"
 [8] "Advanced" "Emerging" "Emerging" "Advanced" "Advanced" "Advanced" "Emerging"
[15] "Emerging" "Emerging" "Emerging" "Advanced" "Emerging
```

Plot bipartite graph of G20 affiliation network

```
    P5 G4 G7 BRICS MITKA DAC OECD Cwth N11
ARG  0  0  0     0     0   0    0    0   0
AUS  0  0  0     0     1   1    1    1   0
BRA  0  1  0     1     0   0    0    0   0
CAN  0  0  1     0     0   1    1    1   0
CHN  1  0  0     1     0   0    0    0   0
DEU  0  1  1     0     0   1    1    0   0
FRA  1  0  1     0     0   1    1    0   0
GBR  1  0  1     0     0   1    1    1   0
IDN  0  0  0     0     1   0    0    0   1
IND  0  1  0     1     0   0    0    1   0
ITA  0  0  1     0     0   1    1    0   0
JPN  0  1  1     0     0   1    1    0   0
KOR  0  0  0     0     1   1    1    0   1
MEX  0  0  0     0     1   0    1    0   1
RUS  1  0  0     1     0   0    0    0   0
SAU  0  0  0     0     0   0    0    0   0
TUR  0  0  0     0     1   0    1    0   1
USA  1  0  1     0     0   1    1    0   0
ZAF  0  0  0     1     0   0    0    1   0
```

```
# bipartite graph with a vertical layout with clustering information
bmgraph(G20, layout = "bipc", clu = list(ac, ec), cex = 4)
```

- Plot as a “clustered” bipartite graph
- Clustering information is given as a list since it is for two domains

Constructing G20 affiliation network with bridges

```
# Option: P5 G4 MITKA none
acb <- factor(ac, levels = c("P5", "G4", "MITKA", "none"))
acb[which(G20[,1]==1)] <- "P5"    ; acb[which(G20[,2]==1)] <- "G4"
acb[which(G20[,5]==1)] <- "MITKA"; acb[which(is.na(acb))] <- "none"
```

```
[1] none MITKA G4 none P5 G4 P5 P5 MITKA G4 none G4 MITKA MITKA P5 none MITKA P5 none
Levels: P5 G4 MITKA none
```

```
# bridge organisations
bridges <- which(acb!="none")
```

```
# extract from G20 only countries affiliated to bridge organisations
G20B <- G20[bridges, c(1:5)]
```

Plot binomial projection of bridged network (isomorphic to 2nd bipartite graph)

```
bmgraph(G20B, layout = "force", seed = 321)
```

Plotting skeleton trade network of G20B

```
# set working directory; see also getwd()
setwd(PATH)

# load G20B skeleton trade network data
load(file = "PATH/G20Bnet.Rdata")

# define scopes and clustering information
club <- c(3, 2, 2, 1, 1, 1, 3, 2, 1, 3, 3, 2, 3, 1)
scp <- list(cex = 4, pos = 0, fsize = 8, fstyle = "bold")
scpb <- list(vcol = c("#BCEE68","#BDB76B","#66CDAA"),
             ecol = c("#00BFFF","#CD5B45"))
```

```
# plot graph combining different types of scopes
multigraph(G20Bnet, valued = TRUE, scope = c(scp, scpb), clu = club)
```

Multilevel systems

addressing complexity

A *multilevel network* X^M for vertex sets N (domain), M (codomain), and edge sets E

$$X^M = \langle\, N,\ M,\ E_N,\ E_M,\ E_{N\times M} \,\rangle$$

- An *affiliation* network is $X^B = \langle N, M, E_{N\times M}\rangle$, where $E_N, E_M = \varnothing$
- A *valued* network $X^V = \langle N, E, V\rangle$ with V for weights
- A *multiplex* system adds R for types of E
- A *dynamic* system has $t > 1$ time stamps, $X^+ = X_1 \ldots X_t$

Multilevel structure (?) of G20 with bridges

co-membership

Functions `mlvl()` and `mlgraph()` allow constructing and plotting multilevel structures

```
# co-membership network matrix
G20Bcn <- mlvl(y = G20B, type = "cn")

# plot valued graph of co-membership network (default circular layout)
mlgraph(G20Bcn, valued = TRUE)
```

However, function `multigraph()` supports multilevel networks as well

```
# valued graph with co-membership values
multigraph(G20Bcn, valued = TRUE, values = TRUE, undRecip = TRUE)
```

Multilevel structure of G20 with bridges

actors co-affiliation

```r
# multilevel with co-affiliation of actors
G20Bcn2 <- mlvl(x = G20Bnet, y = G20B, type = "cn2")
```

```
...

$lbs
$lbs$dm
 [1] "AUS" "BRA" "CHN" "DEU" "FRA" "GBR" "IDN" "IND" "JPN" "KOR" "MEX" "RUS" "TUR" "USA"

$lbs$cdm
[1] "P5"    "G4"    "G7"    "BRICS" "MITKA"


$modes
[1] "1M" "1M" "2M"

attr(,"class")
[1] "Multilevel" "cn2"
```

```r
# plot multilevel structure
mlgraph(G20Bcn2)
```

Multilevel structure of G20B with clustering information

actors co-affiliation

Additional clustering information for events and `club` still for actors

```
# clustering information with events (may not needed in future)
club2 <- list(club, rep(1, ncol(G20B)) )
```

```
[[1]]
 [1] 3 2 2 1 1 1 3 2 1 3 3 2 3 1

[[2]]
[1] 1 1 1 1 1
```

```
# plot multilevel network as circular layout and updated clustering info
mlgraph(G20Bcn2, valued = TRUE, scope = c(scp, scpb), clu = club2)
```

Multilevel structure with binomial projection

```
# multilevel with binomial projection
G20Bbp <- mlvl(x = G20Bnet, y = G20B, type = "bpn")
```

```
$mlnet
, , M

      AUS BRA   CHN  DEU  FRA  GBR IDN IND JPN KOR  MEX RUS TUR USA P5 G4 G7 BRICS MITKA
AUS     0   0  5829    0    0    0   0   0   0 412    0   0   0   0  0  0  0     0     0
BRA     0   0     0    0    0    0   0   0   0   0    0   0   0   0  0  0  0     0     0
CHN     0   0     0    0    0    0   0   0   0   0    0   0   0   0  0  0  0     0     0
DEU     0   0 14417    0 7358  928   0   0   0   0    0   0   0   0  0  0  0     0     0
FRA     0   0  6714 7553    0 1345 443   0   0 921    0   0   0 254  0  0  0     0     0
GBR     0   0  1734  641 1170    0   0   0   0   0    0   0   0   0  0  0  0     0     0
IDN     0   0     0    0    0    0   0   0   0   0    0   0   0   0  0  0  0     0     0

...


, , 3

      AUS BRA CHN DEU FRA GBR IDN IND JPN KOR MEX RUS TUR USA P5 G4 G7 BRICS MITKA
AUS     0   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  0     0     1
BRA     0   0   0   0   0   0   0   0   0   0   0   0   0   0  0  1  0     1     0
CHN     0   0   0   0   0   0   0   0   0   0   0   0   0   0  1  0  0     1     0
DEU     0   0   0   0   0   0   0   0   0   0   0   0   0   0  0  1  1     0     0
FRA     0   0   0   0   0   0   0   0   0   0   0   0   0   0  1  0  1     0     0
GBR     0   0   0   0   0   0   0   0   0   0   0   0   0   0  1  0  1     0     0
IDN     0   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  0     0     1

...
```

Multilevel structure with binomial projection

scope and clustering

Define additional scopes to handle events

```
# shapes and color of vertices for actors and events
scpm <- list(ecol = c("blue","red","orange"), pch = c(21,15), vcol0 = 8,
             vcol = c("#BCEE68","#BDB76B","#66CDAA","#838B8B","#FF7F00"))

# four classes of actors as factor with explicit levels
acc <- factor(ac, levels = c("A-G7", "Advanced", "E-BRICS", "Emerging"))
acc[which(G20[,3]==1)] <- "A-G7"; acc[which(G20[,4]==1)] <- "E-BRICS"

# clustering information for multilevel structure
cluml <- list(acc[bridges], rep(1, nrow(G20B)))
```

```
[[1]]
 [1] Advanced E-BRICS  E-BRICS  A-G7     A-G7     A-G7     Emerging E-BRICS  A-G7     Advanced
[11] Emerging E-BRICS  Emerging A-G7
Levels: A-G7 Advanced E-BRICS Emerging

[[2]]
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Multilevel structure with binomial projection

plotting

Multilevel network with binomial projection (updated clustering information)

```
# plot with default circular layout and recycling scope
mlgraph(G20Bbp, scope = c(scp, scpm), clu = cluml)
```

```
# clustering information of the two domains in 'G20Bbp'
nr <- c(rep(1,nrow(G20B)), rep(2,ncol(G20B)))
```

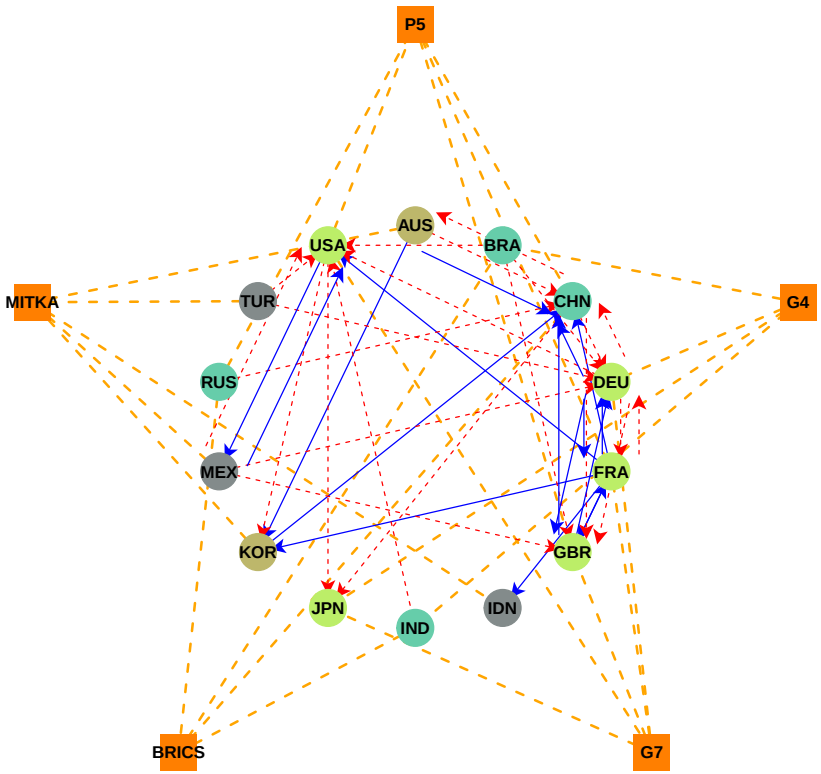
```
# [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
```

Plot multilevel graph with binomial projection

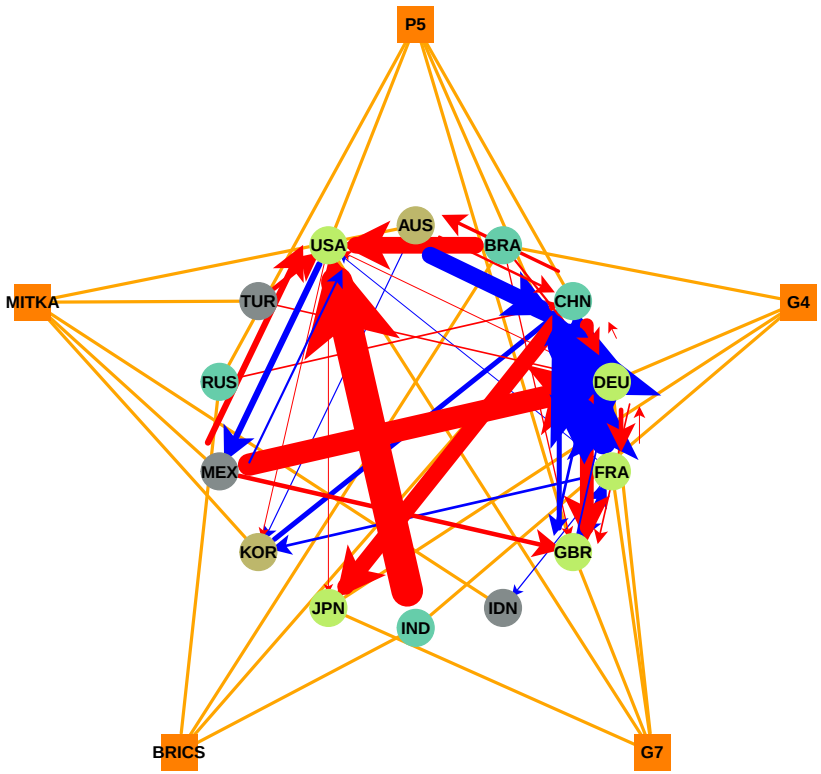
```
# concentric layout with two radii and recycled scopes
mlgraph(G20Bbp, layout = "conc", nr = nr, scope = c(scp,scpm), clu = cluml)
mlgraph(G20Bbp, layout = "conc", nr = nr, scope = c(scp,scpm), clu = cluml,
        valued = TRUE)
```

Multilevel structure with binomial projection

concentric layout



valued



Positional analysis of the Multilevel structure

Functions `reduc()` to reduce array structures

```
# positional system actors with clustering information
PSG20Ba <- reduc(G20Bnet, valued = TRUE, lbs = c("G7.C","BRICS.C","MITKA.C"),
                 clu = club)
```

```
, , M

         G7.C BRICS.C MITKA.C
G7.C    19249   22865    3538
BRICS.C     0       0       0
MITKA.C   752    7572     412

, , H

         G7.C BRICS.C MITKA.C
G7.C     3050     296     293
BRICS.C 29577     584    1187
MITKA.C 13477     860       0
```

```
# Positional system events ('row' option for two-mode networks)
PSG20Be <- reduc(G20B, valued = TRUE, lbs = c("G7.C","BRICS.C","MITKA.C"),
                 clu = club, row = TRUE)
```

Multilevel structure of positional system for G20B

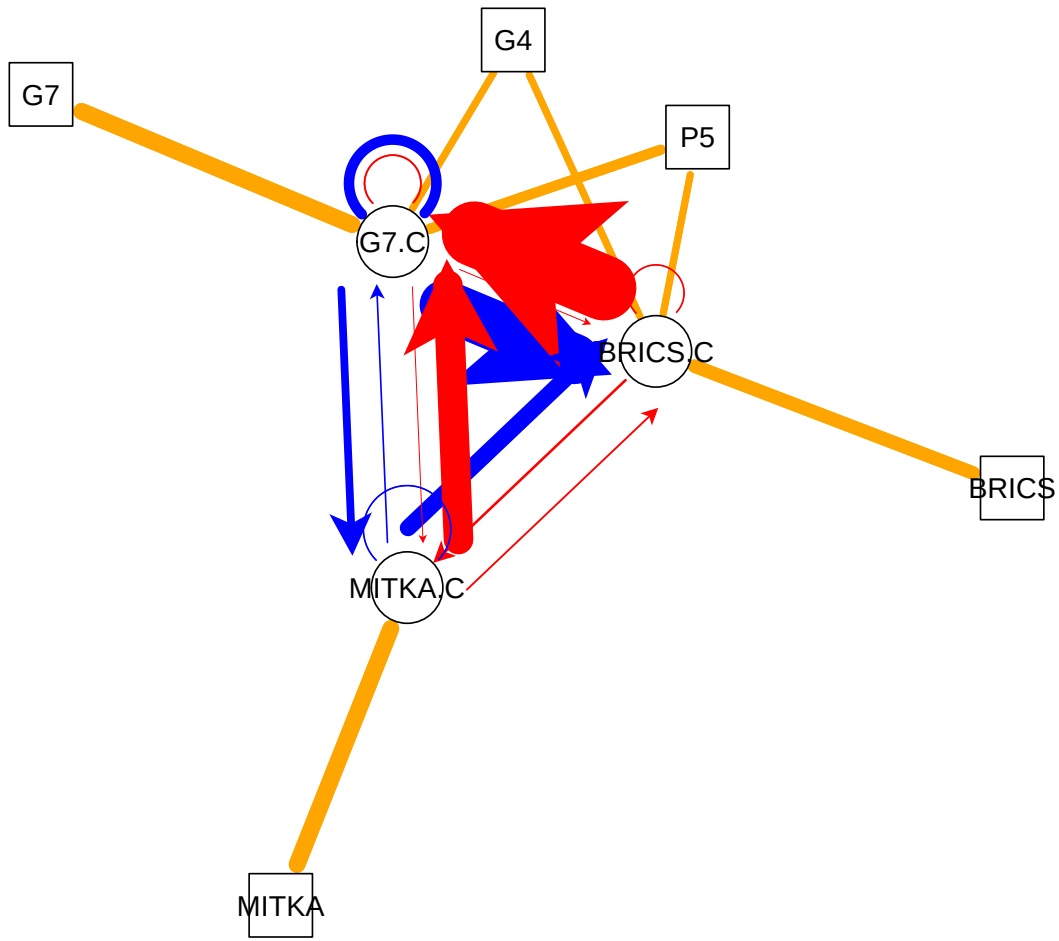
```
# multilevel structure with binomial projection and symmetric co-domain
PSG20Bbp <- mlvl(x = PSG20Ba, y = PSG20Be, type = "bpn", symCdm = TRUE)
```

```
$mlnet
, , m

         G7.C BRICS.C MITKA.C P5 G4 G7 BRICS MITKA
G7.C    19249   22865    3538  0  0  0     0     0
BRICS.C     0       0       0  0  0  0     0     0
MITKA.C   752    7572     412  0  0  0     0     0
P5          0       0       0  0  0  0     0     0
G4          0       0       0  0  0  0     0     0

 ...

, , 3

        G7.C BRICS.C MITKA.C P5 G4 G7 BRICS MITKA
G7.C       0       0       0  3  2  5     0     0
BRICS.C    0       0       0  2  2  0     4     0
MITKA.C    0       0       0  0  0  0     0     5
P5         3       2       0  0  0  0     0     0
G4         2       2       0  0  0  0     0     0
G7         5       0       0  0  0  0     0     0
BRICS      0       4       0  0  0  0     0     0
MITKA      0       0       5  0  0  0     0     0

...
```

Valued multilevel structure of positional system for G20B

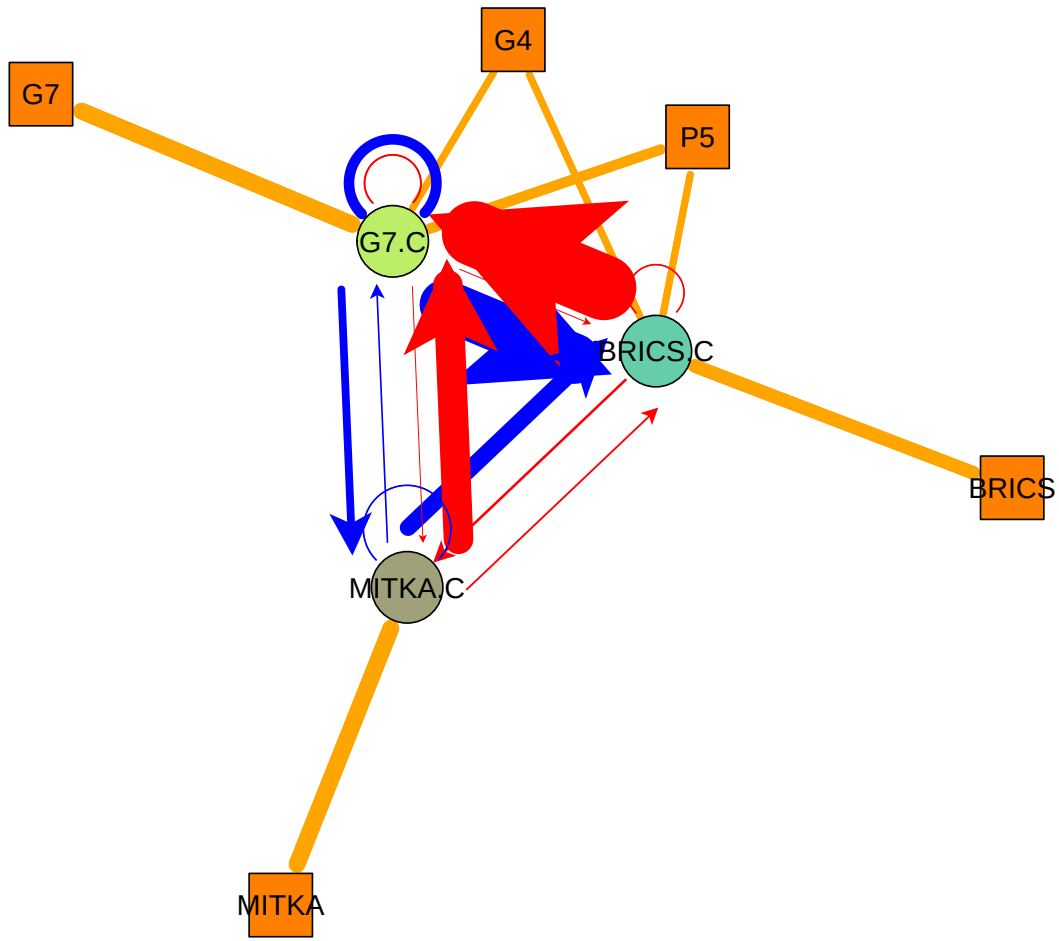
edge colors are for milk, honey, and affiliation ties

```
# plot multilevel graph of positional system
mlgraph(PSG20Bbp, valued = TRUE, layout = "force", seed = 1, cex = 6, pos = 0,
        ecol = c("blue","red","orange"), fsize = 11)
```



Valued multilevel structure of positional system for G20B

```
# define scopes multilevel positional system and plot
scpps <- list(cex=6, ecol=c("blue","red","orange"), pos=0, fsize=11)
scpps2 <- list(vcol=c("#BCEE68","#66CDAA","#A0A17B","#FF7F00"),
               clu=c(1:3, rep(4,5)))
mlgraph(PSG20Bbp, layout="force", seed=1, valued=TRUE, scope=c(scpps,scpps2))
```



4. Network analysis in history and archaeology

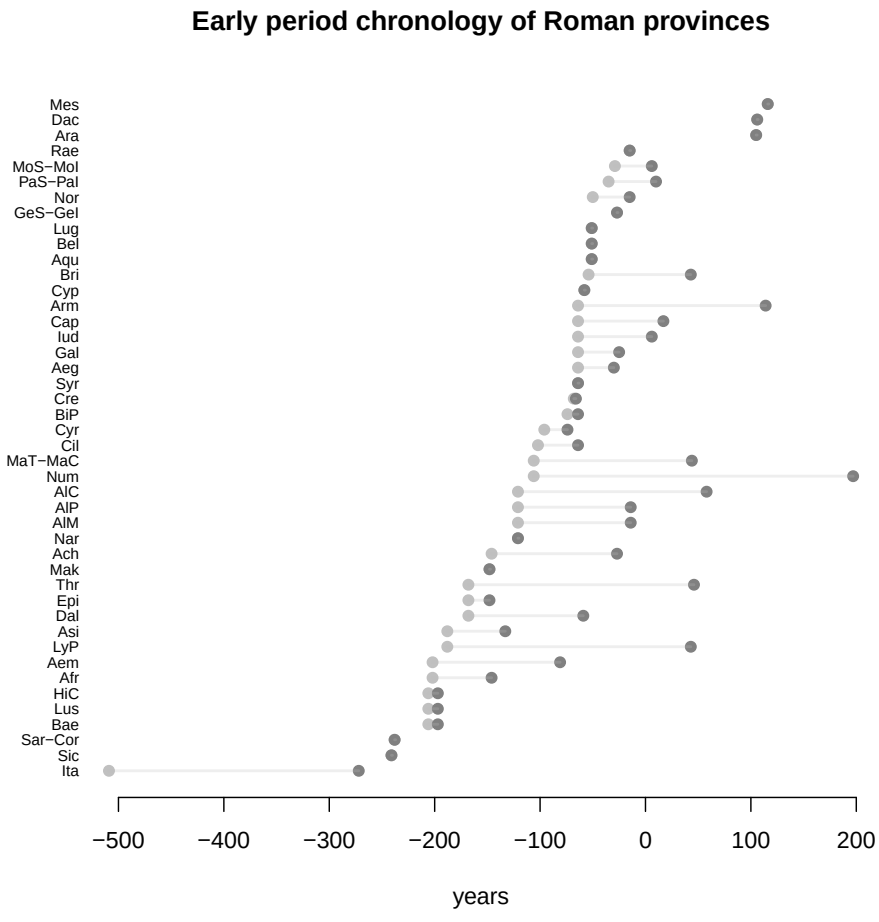
Work-in-progress: Ancient Roman world

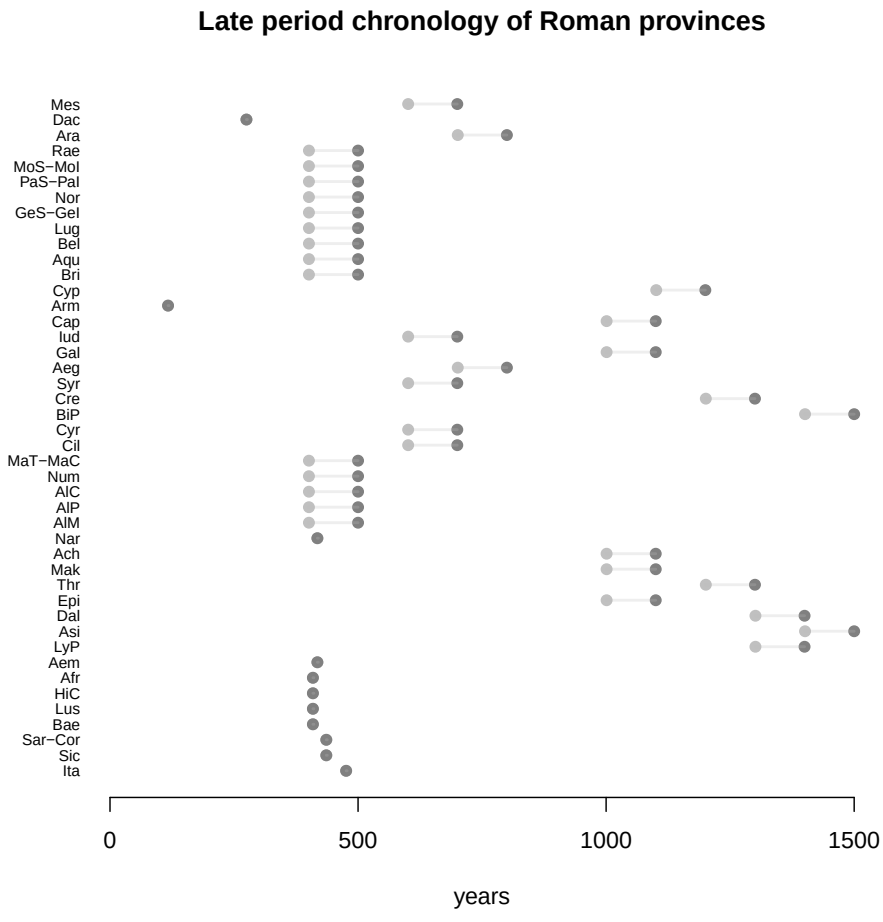
Roman Provinces (117 AD)

i Aegyptus
ii Alpes Cottiae
iii Alpes Maritimae
iv Alpes Poenninae
v Armenia
vi Assyria
vii Britannia
viii Cilicia
ix Dacia
x Dalmatia
xi Galatia
xii Gallia Aquitania
xiii Gallia Belgica
xiv Gallia Lugdunensis
xv Germania Inferior
xvi Germania Superior
xvii Hispania Tarraconensis
xviii Judaea
xix Lusitania
xx Moesia
xxi Noricum
xxii Pannonia
xxiii Raetia
xxiv Syria
xxv Thracia

Ancient Roman provinces historical periods

www.unrv.com/provinces/

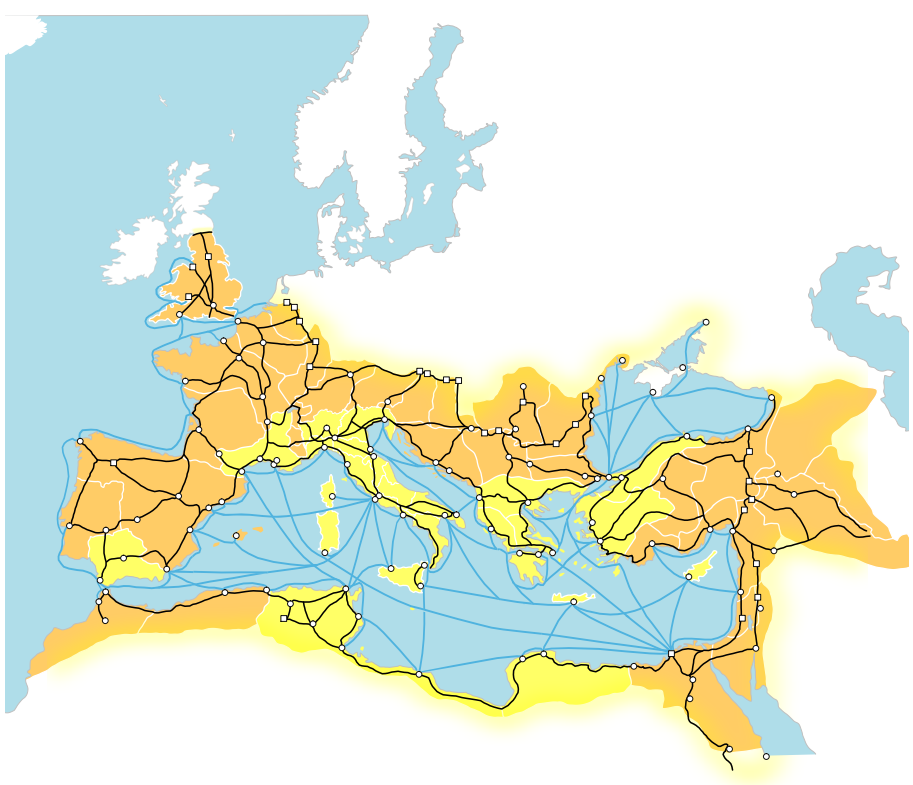


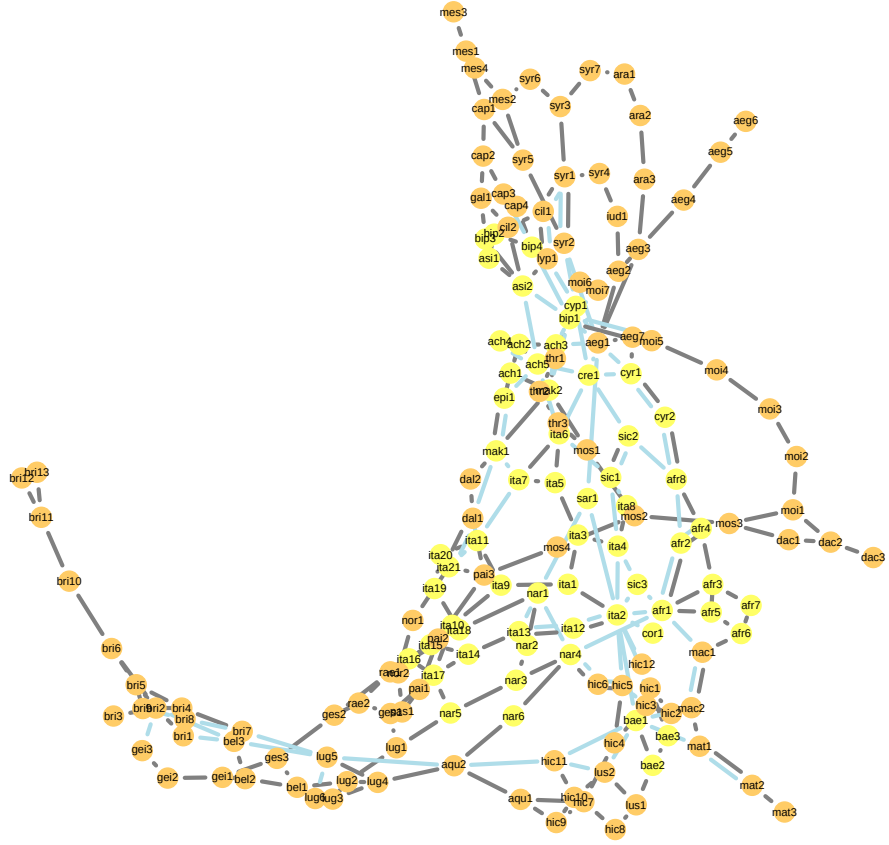



```
sdam::plot.dates(x, taq, tpq, ...)
```

Roman Empire transport network (maximum extent ca. 117 AD)

main roads & maritime routes and province types





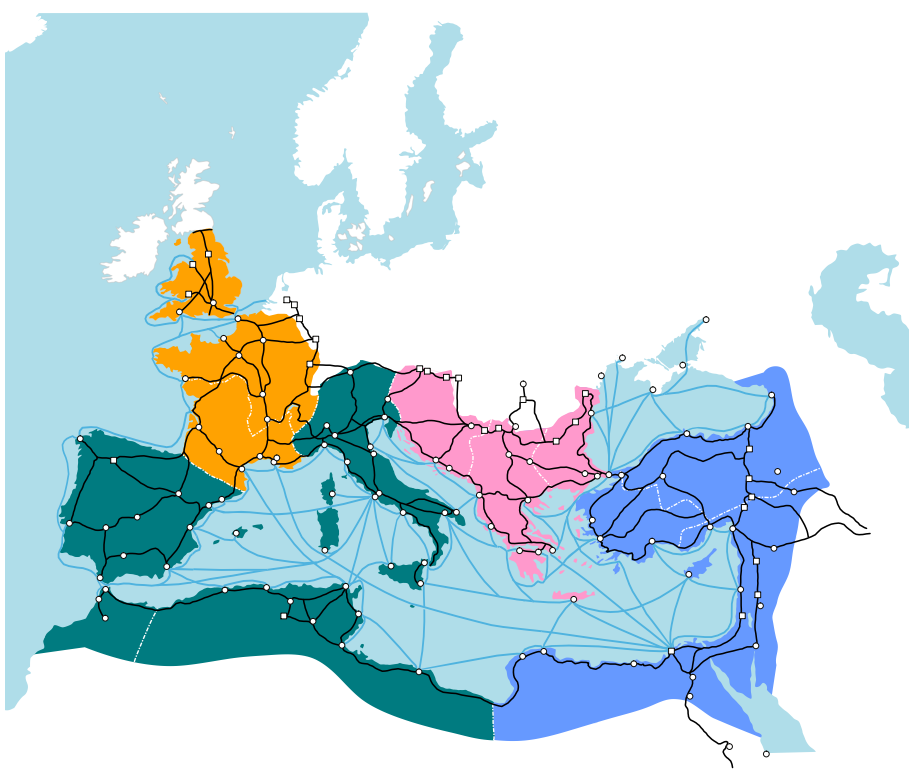
routes based on Rodrigue (2013)

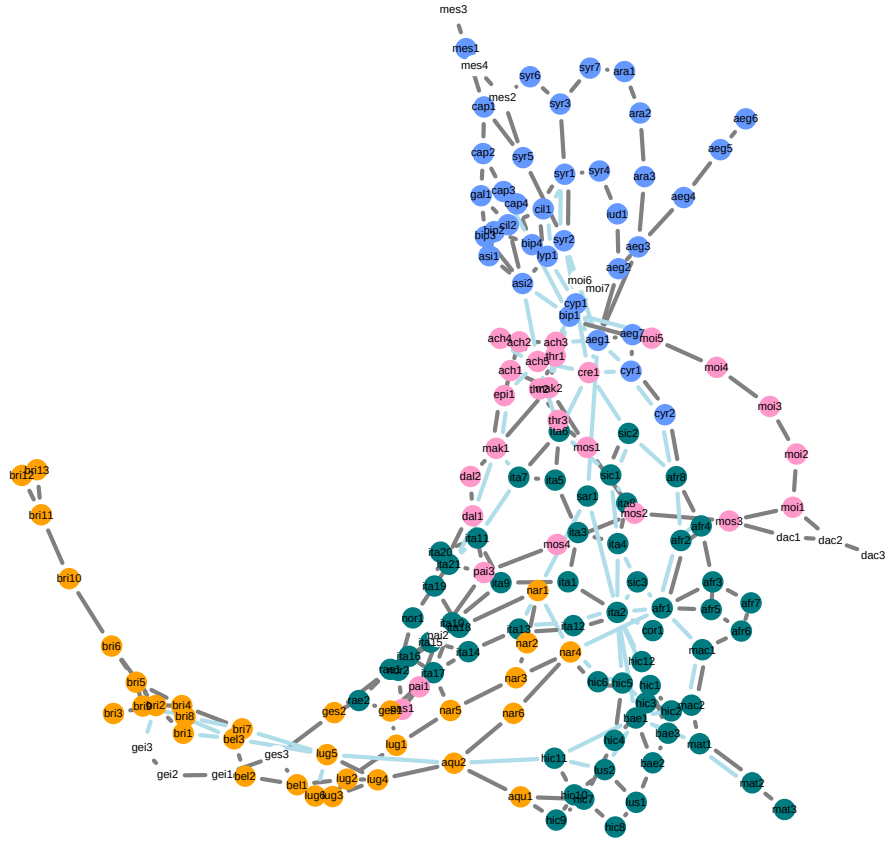
```
sdam::plot.map(type = "si")
```

senatorial , imperial · maritime , roads

Roman Empire transport network

main roads & maritime routes under a tetrarchy (ca. 284 AD)



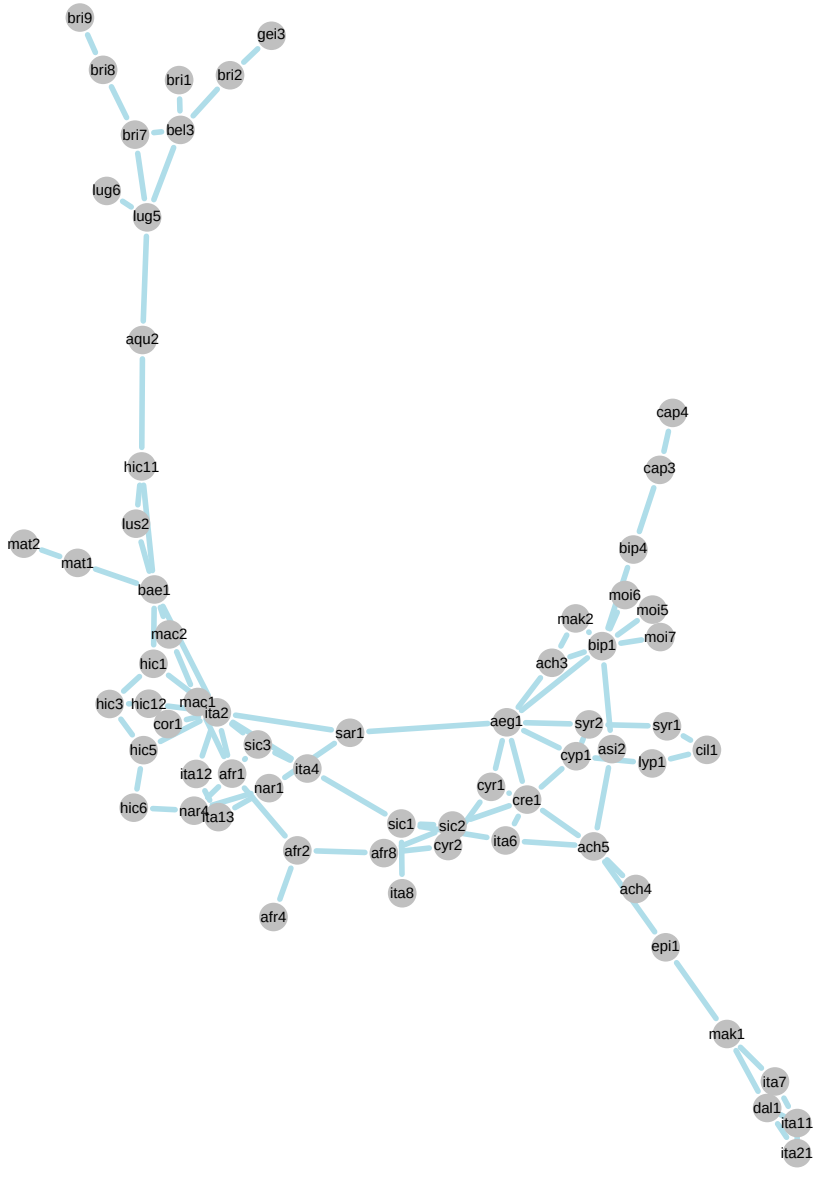


```
sdam::plot.map(type = "tetra")
```

MAXIMIAM , DIOCLECIANO · GALERIO , CONSTANCIO

Main maritime routes

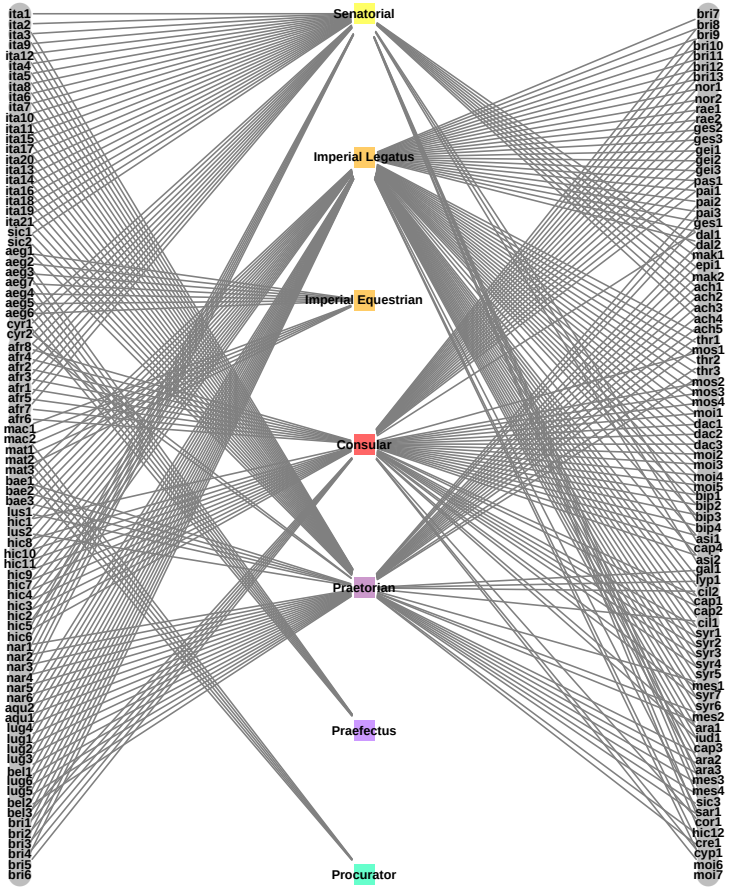
Roman Empire transport network



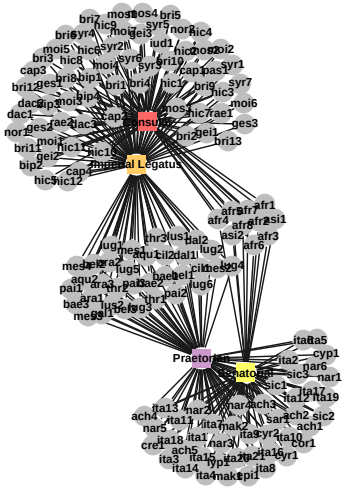
Nodes with province and government types

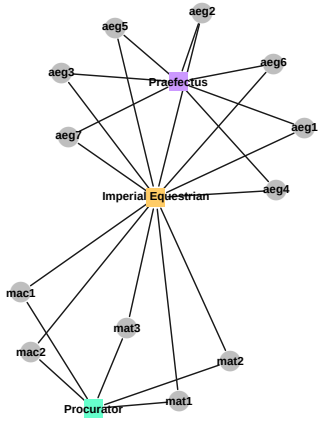
"multimodal" network graphs

Bipartite



Binomial projection

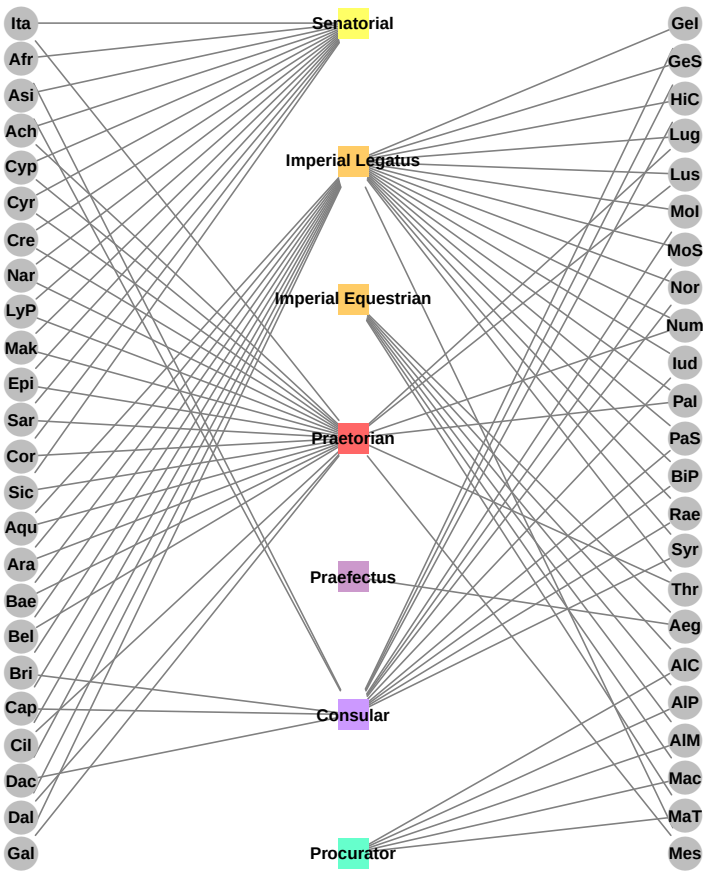




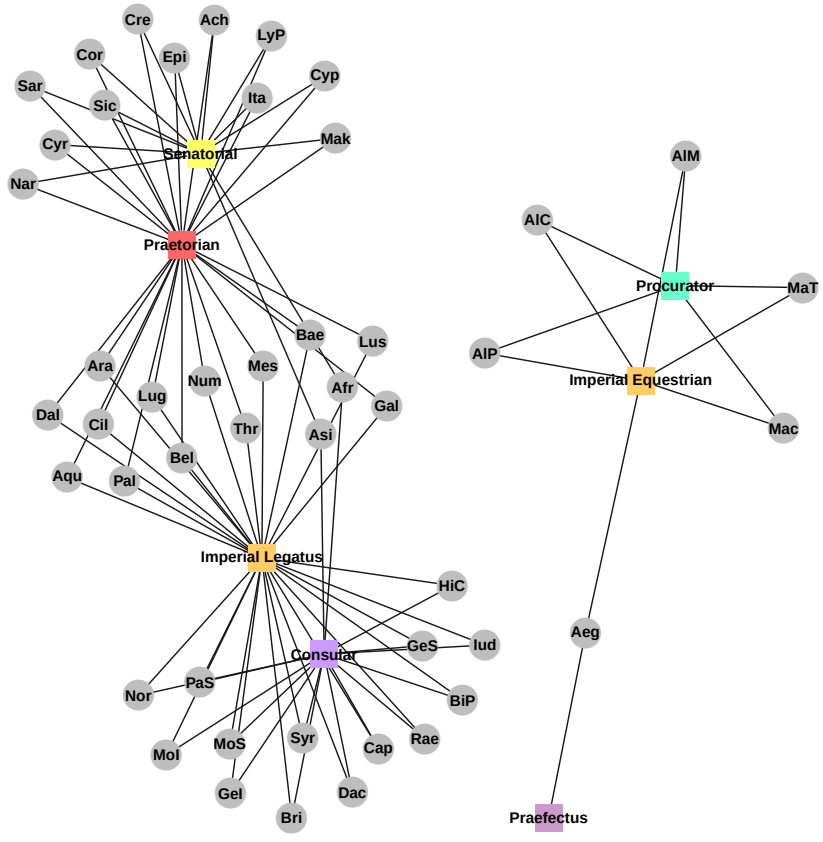
Roman provinces and government types

multimodal network graphs

Bipartite

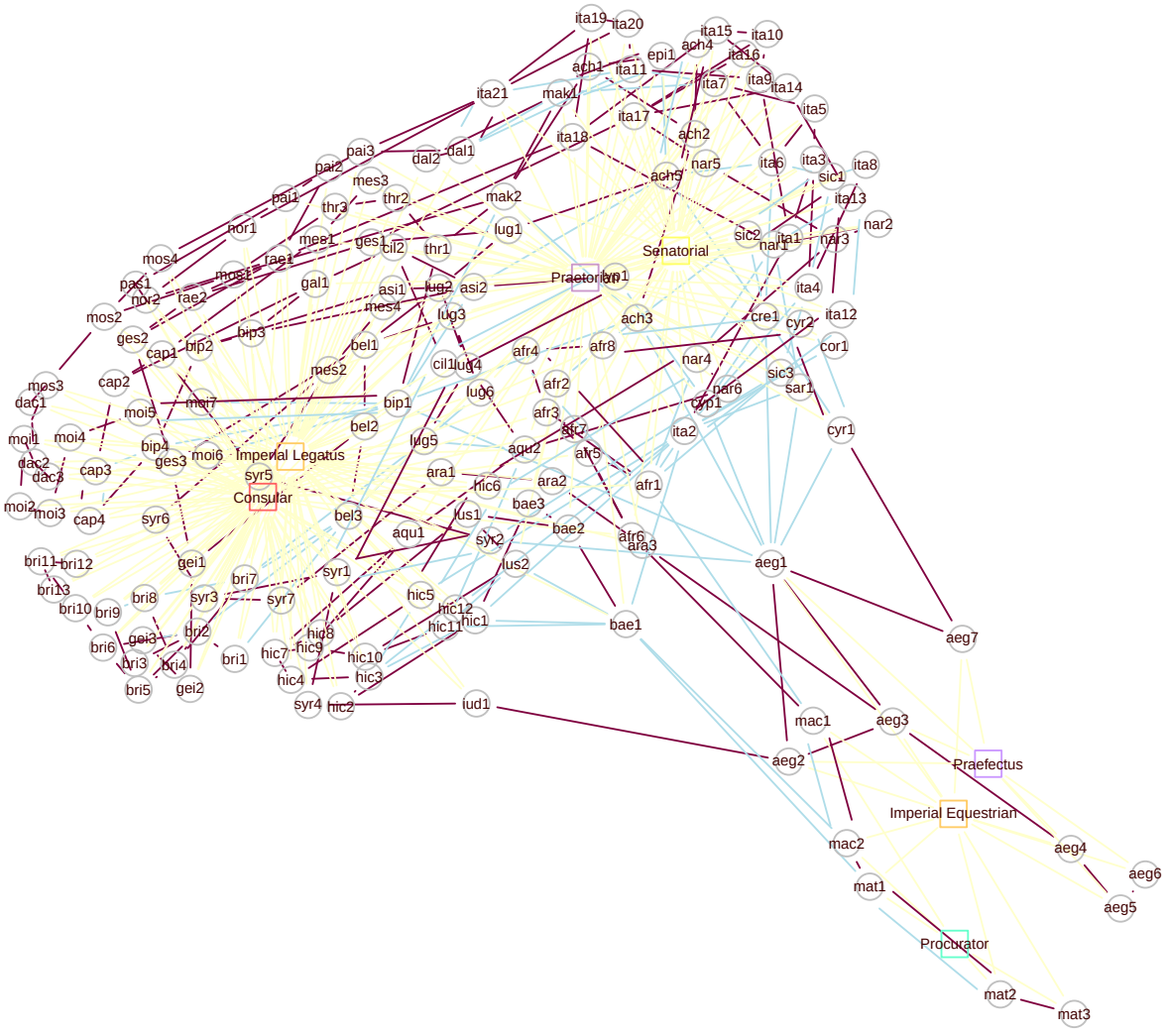


Binomial projection



Multilevel structures

a projection of Roman Empire transport network with political affiliations

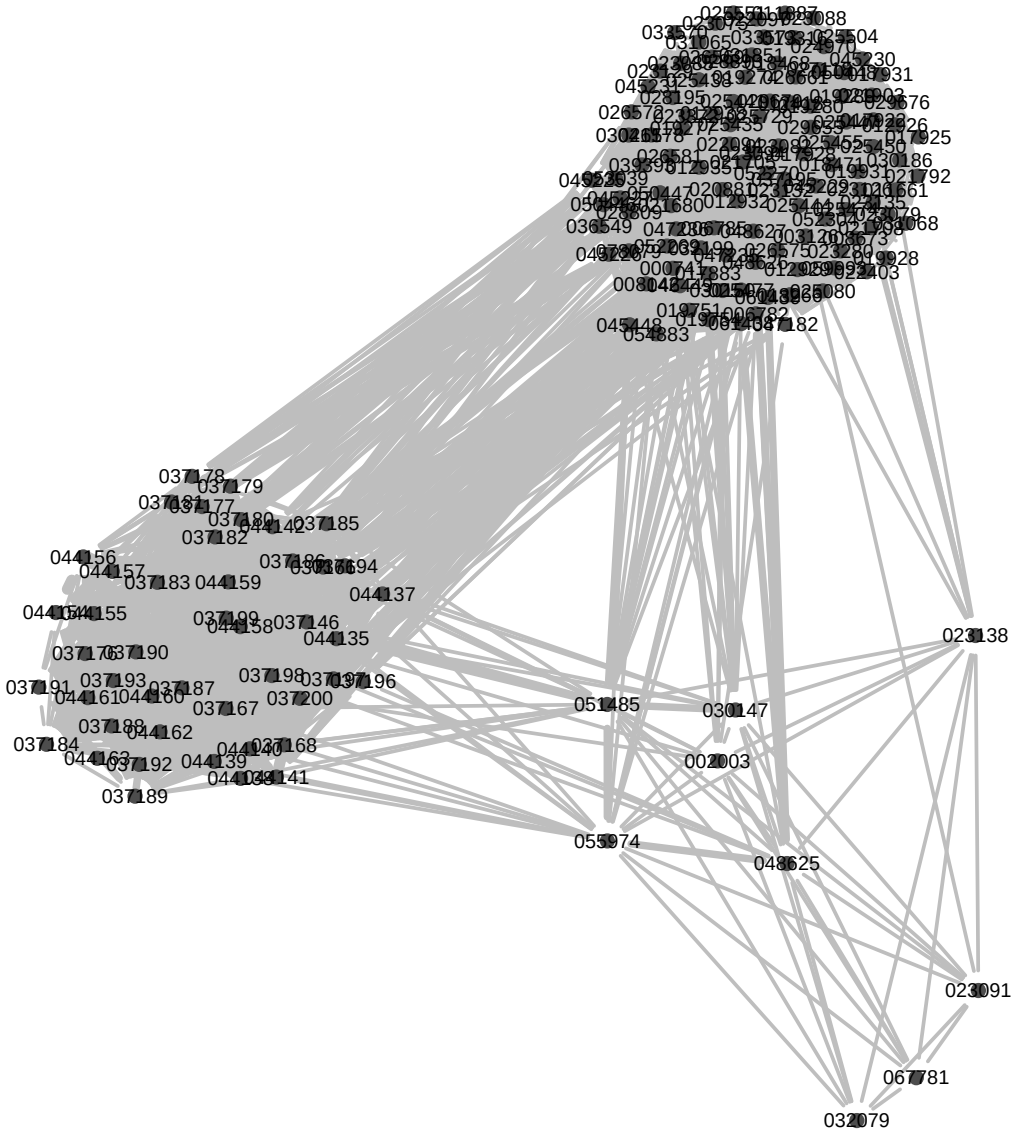


circles: *actors* in N

squares: *events* in M

Epigraphic networks

addressing social complexity



EDH dataset with attribute simple matching

(cf. SDAM project)

Data sets from variable attributes similarities

sdam **R** package

```
data("EDH")
```

```
# for imputation
dates = c("not_before", "not_after")
# characteristics
vars = c("findspot_ancient", "type_of_inscription", "type_of_monument", "language")
```

- edhv is a data frame that records the chosen variables from EDH.

```
# EDH dataset as a dataframe with variables
edhv = sdam::edhw(vars=c(dates, vars, "province_label"), as="df")
```

```
aeg = sdam::edhw(x=edhv, vars=c(dates, vars), province="Aeg")
```

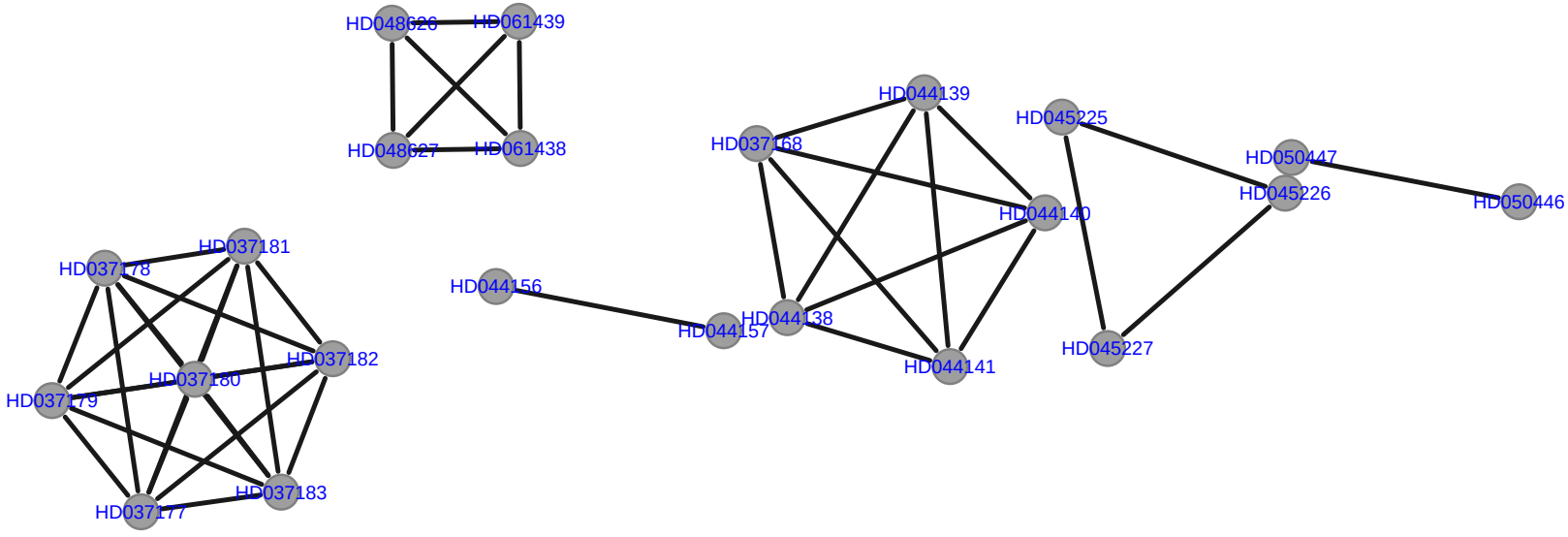
```
           id     type_of_monument        type_of_inscription not_before not_after    language
741  HD000741                cliff identification inscription       0001      0200       Latin
2003 HD002003 architectural member identification inscription      -0116      <NA> Greek-Latin
3126 HD003126              diptych                       <NA>       0101      0200       Latin
6782 HD006782                 <NA>                       <NA>       0212      0250       Latin
6785 HD006785                 <NA>                       <NA>       0200      0250       Latin
8142 HD008142               tabula                    epitaph       0065      <NA>       Latin
```

(extract)

```
 Dal  GeS  HiC  Bri  Afr  PaS  Rom  Dac  GeI  PaI  Bae  Nor  Num  LaC  MoI  Bel  Lus  MoS  Mak  Nar
7711 7024 5221 4561 4496 4384 4331 3682 3352 3240 3035 2789 2642 2606 2073 1691 1577 1515 1409 1403
 Ach  VeH  MaC  Rae  ApC  Etr  Sam  Lug  Asi  AlC  Aqu  AlM  Thr  Syr  Umb  BiP  MaT  BrL  Ara  Aem
1328 1161 1127 1017  760  671  648  597  491  476  427  416  411  403  348  293  291  246  234  231
 Sar  Gal  AlP  Sic  Iud  Bar  Tra  Aeg  Pic  Lig  Epi  Cyr  Cil  ReB  AlG  Cre  Cap  LyP  Cor  Cyp
 229  211  206  193  192  189  180  172  170  153  144   90   86   78   77   66   63   59   46   35
 Mes  Arm  Ass  Inc  Tri  Val
  12    3    0    0    0    0
```

Aegyptus: Network cliques with all common variables

findspot ancient, type of inscription, type of monument, language




```
sdam::edhw(aeg, id=component1)
sdam::edhw(aeg, id=component2)
```

```
           id type_of_monument type_of_inscription not_before not_after language findspot_ancient
39521 HD037168            stele             epitaph       0110      0129    Greek      Terenouthis
46474 HD044138            stele             epitaph      -0030      0100    Greek      Terenouthis
46475 HD044139            stele             epitaph       0156      0179    Greek      Terenouthis
46476 HD044140            stele             epitaph      -0011      <NA>    Greek      Terenouthis
46477 HD044141            stele             epitaph      -0026      <NA>    Greek      Terenouthis
```

```
           id type_of_monument type_of_inscription not_before not_after language findspot_ancient
46492 HD044156            block  votive inscription       <NA>      <NA>    Greek           Kellis
46493 HD044157            block  votive inscription       <NA>      <NA>    Greek           Kellis
```

References


Sampson, FS *A novitiate in a period of change: An experimental and case study of social relationships*, Cornell Univ. 1969.


White, HC *An Anatomy of Kinship: Mathematical Models for Structures of Cumulated Roles*. Prentice-Hall. 1963

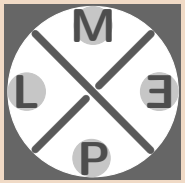

Pattison, P *Algebraic Models for Social Networks*. Cambridge Univ. Press. 1993


Ostoic, JAR *Algebraic Analysis of Social Networks*. Wiley, 2021


Ostoic, JAR 'Algebraic Analysis of Multiple Social Networks with multiplex,' *Journal of Statistical Software* **92**(11), 1–41. 2020


R Development Core Team, ***R**: A language and environment for statistical computing*, version 4.1.1


SDAM: Social Dynamics in the Ancient Mediterranean project `http://sdam.au.dk`



github.com/mplex

jaro@cas.au.dk