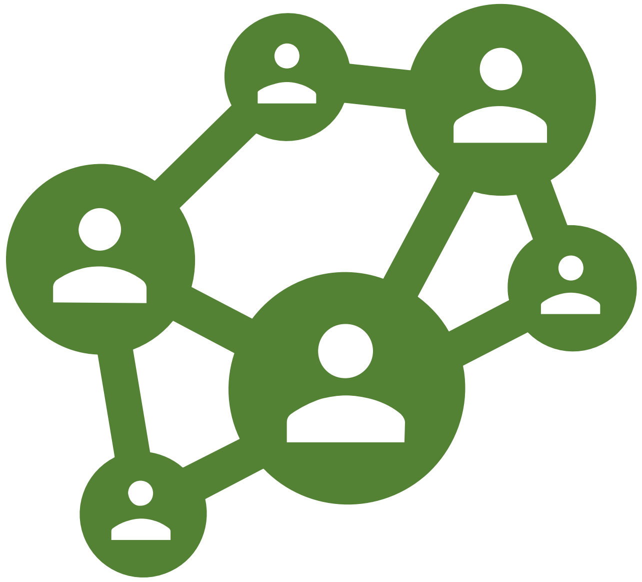
Social Network Analysis
Day 1

Advanced Winter School in Computational Social Science 2025

Carl Nordlund
Maël Lecoursonnais
Kazuki Sakamoto



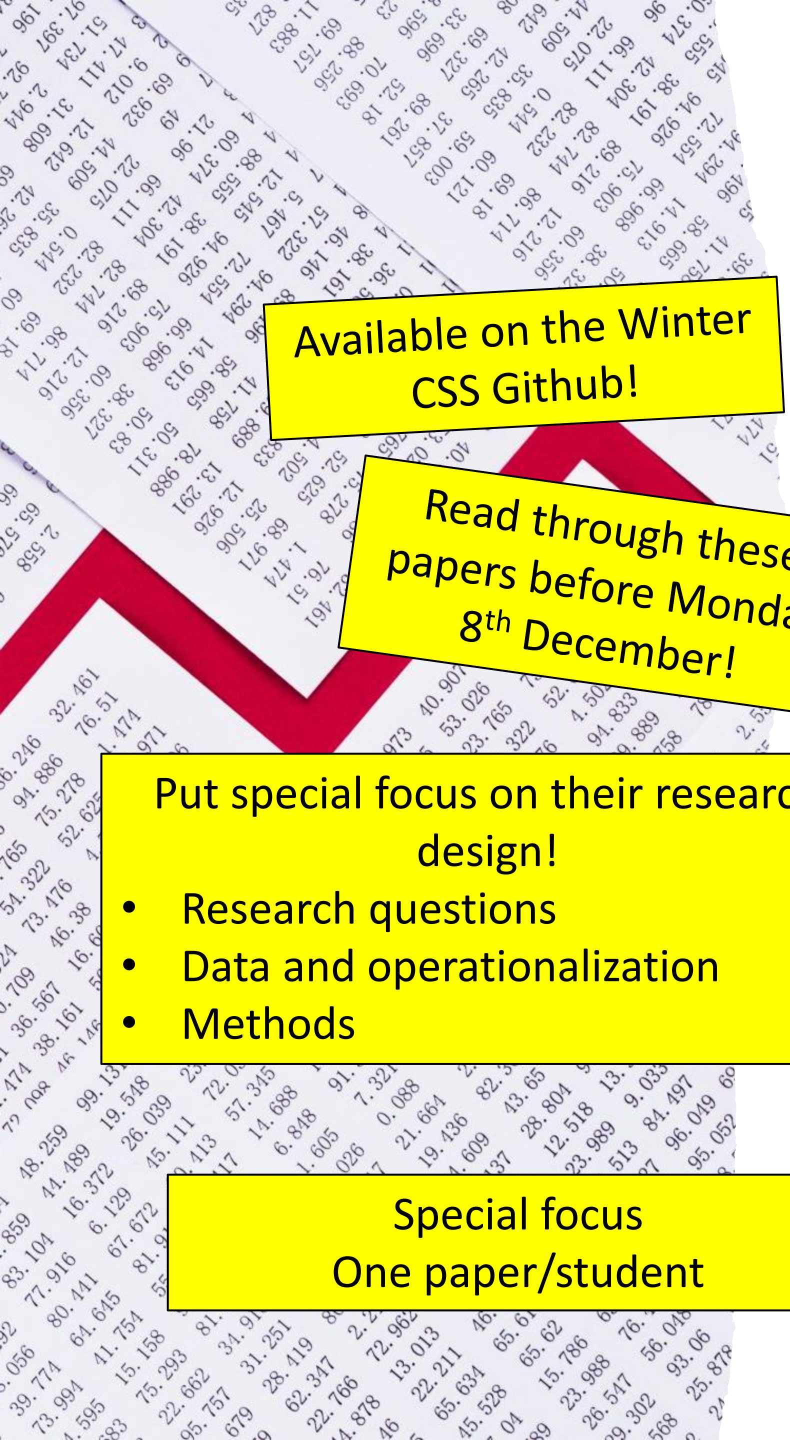
Day 1: Friday 5th Dec

- Introduction to module
- Relational thinking and research design
- Communities and other cohesive subgroups
- Role structures and blockmodeling
- Lab details

Day 2: Monday 8th Dec

- Comparing networks (QAP)
- Network null models: rewiring, configuration models, backboning
- Simulations
- Relations and nodal attributes: introduction to network dynamics
- Research design considerations

Social Network Research Design component



Available on the Winter CSS Github!

Read through these papers before Monday 8th December!

Put special focus on their research design!
- Research questions
- Data and operationalization
- Methods

Special focus
One paper/student

Three papers

- Gerbrands P, Unger B, Getzner M, et al. (2022) The effect of anti-money laundering policies: an empirical network analysis. *EPJ Data Sci*. 11, 15. https://doi.org/10.1140/epjds/s13688-022-00328-8
- Thiemann C, Theis F, Grady D, Brune R, Brockmann D (2010) The Structure of Borders in a Small World. *PLoS ONE* 5(11): e15422. https://doi.org/10.1371/journal.pone.0015422
- Yang Y, Chawla N V, Uzzi B (2019) A network's gender composition and communication pattern predict women's leadership success. *PNAS*, 116(6), pp. 2033-2038. https://www.pnas.org/cgi/doi/10.1073/pnas.1721438116

Paper assignments

Name	Paper
Great Uchechukwu Udochi	Gerbrands et al
Yevhen Voronin	Thiemann et al
Lu Yan	Yang et al
Ratna Aini Hadi	Gerbrands et al
Carlos Villalobos	Thiemann et al
Clara Englert	Yang et al
Bulin Li	Gerbrands et al
Eric Ngo	Thiemann et al
Leon Klingborg	Yang et al
Jasper Darin	Gerbrands et al
Landry Botokou	Thiemann et al
Milos Milovanovic	Yang et al
Florencia Piñeyrúa	Gerbrands et al

Aim and Purpose

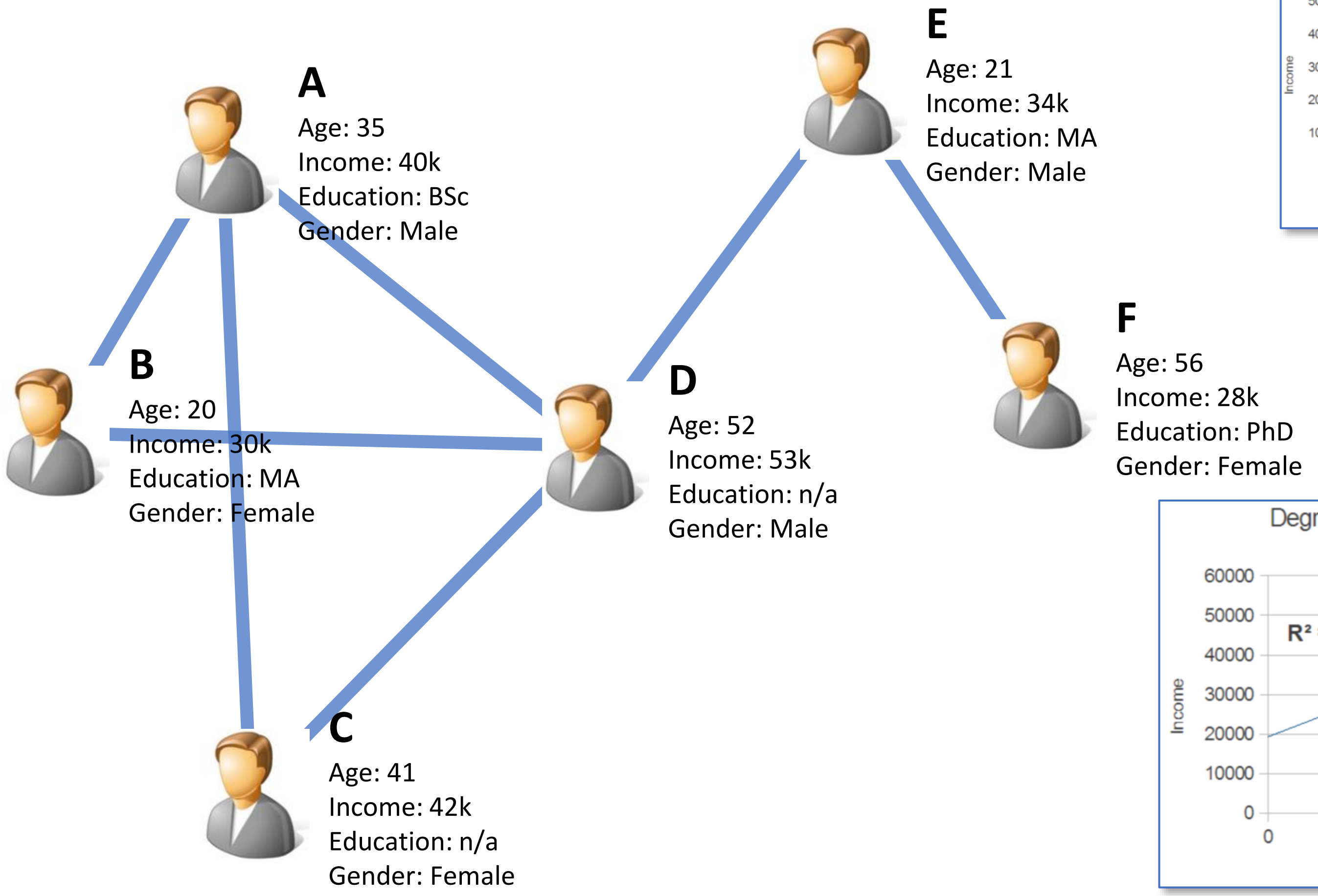
- Introduction to network thinking
- Get gears running
  - Seeing networks
  - Foundational tools/starters
  - Pave the way forward (RQs, more advanced methods)
- Limitations
  - Only 2 days
  - First principles assumed
  - More structural (less attributional)
  - Much more in the SNA toolbox

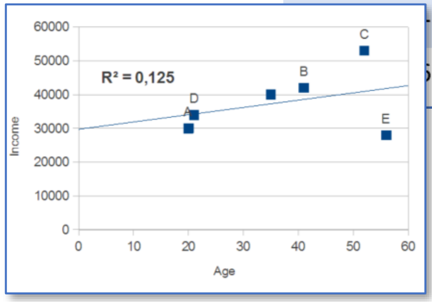
Some repetition and recognition from CSS Summer school

Relational vs attributional data

- Entities: actors / nodes / vertices (here: social ones, e.g. individuals)
- Actor attributes ("traditional" data table)
- Relations: data *between* entities (e.g. friendship, kinship, shared affiliations)

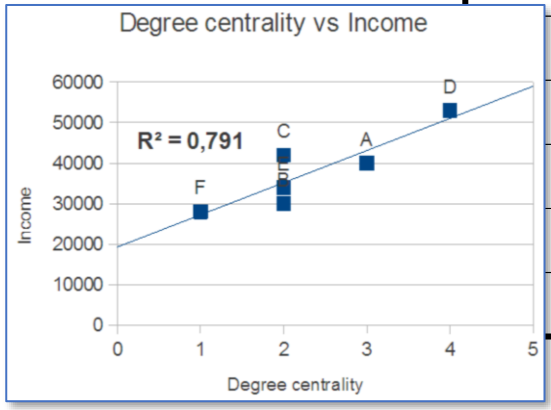
Label	Age	Income	Education	Gender
A	35	40k	BSc	Male
B	20	30k	MA	Female
C	41	42k	n/a	Female
D	52	53k	n/a	Male
		34k	MA	Male
		28k		





Friendship (reciprocated) [matrix-format]

	A	B	C	D	E	F
A	-	1	1	1	0	0
		-	0	1	0	0
		0	-	1	0	0
		1	1	-	1	0
		0	0	1	-	1
		0	0	0	1	-



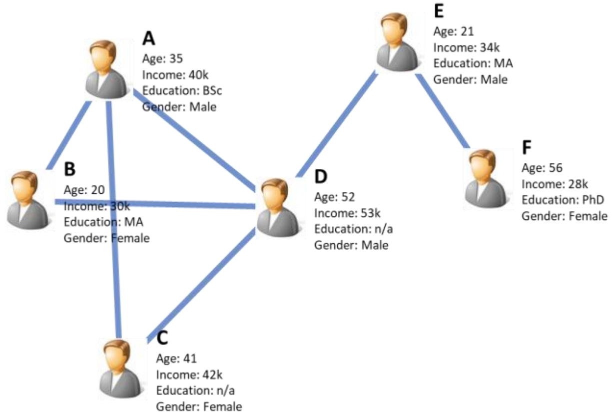
Relational data

- Capturing the in-between
- Inter-dependence of observations
  - Removing a ‘data point’ can have grand implications
- Sampling typically problematic
  - System boundaries for full populations important
- Supplementing ‘conventional’ (attributional) data

Ethical considerations!

Friendship (reciprocated) [matrix-format]

	A	B	C	D	E	F
A	-	1	1	1	0	0
B	1	-	0	1	0	0
C	1	0	-	1	0	0
D	1	1	1	-	1	0
E	0	0	0	1	-	1
F	0	0	0	0	1	-



The Robins reminders

- Incorporating network data and metrics far more than a mere methodological extension
- Explicit theoretical commitments about structure and dependence: *that the in-between matters*!
- Going beyond *Hobbesession* with individual, assumedly independent entities to understand social world

Understanding high-school student performance

Classes of high-school students

Understanding drivers behind performance

Variables (attributes)

- Grades, test scores
- Gender
- Ethnicity
- Parents’ education, SES
- Previous grades
- Eye sight (!)

Relational data

- Friendship (self-reported)
- Social interaction (observed)
- Antagonism
- Kinship (siblings)
- Note-borrowing
- Residential neighbors

Affiliation (2-mode) data

- Student clubs/associations
- Sports/hobbies
- Spoken languages

National economic development

Nations as units

- Population
- GDP/capita
- Gini coefficients
- Sectorial employment
- Value-added (creation, trade-in)
- Main exports/imports
- Trade agreements
- Organizational memberships
- Labor demographics
- R & D expenditure

Relational data

- Bilateral trade flows (aggregated)
- Bilateral trade flows (detailed)
- Geographical contiguity
- Trade envoys
- Sanctions, boycotts and military conflicts
- Cultural exchange
- Airline traffic, other infrastructure

Affiliation (2-mode) data

- Trade/treaty agreements
- Organizational memberships

Becoming a (soccer) hooligan

Individuals as units

- Gender
- Age
- Ethnicity
- Interests
- School grades
- Substance abuse
- Family situation
- Employment
- Criminal records

Relational data

- Social exposure
- Peer influence
- Friendships (various domains)
- Rivalry and antagonism

Affiliation (2-mode) data

- Trade/treaty agreements
- Organizational memberships

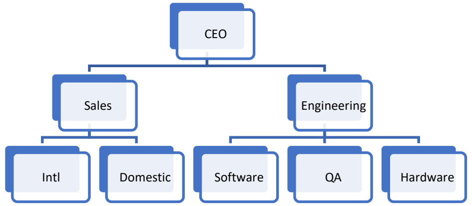
Employee productivity

Individuals as units

- Age
- Education
- Job title
- Salary
- Health care plan
- Tasks completed
- Evaluations
- Skills

Relational data

- Organizational structure (formal)
- Organizational structure (actual!)
- Information flows (individuals and units)
- Inter-personal ties
    - Friendship
    - Antagonism
    - Reports-to
    - Advice-seeking



Popular Norrköping museums

Venues

- Daily visitors
- Type (art, history, contemporary, gallery)
- Entry cost
- Opening hours
- Popularity
- Time spent
- Media mentions
- Online reviews

Relational data

- Road network (betweenness centrality etc.)
- Mobility flows (“bilateral” pedestrians)
- Collaborations with other venues

Affiliation (2-mode) data

- Co-visits (individuals vs. venue)
- Media mentions (as 2-mode)

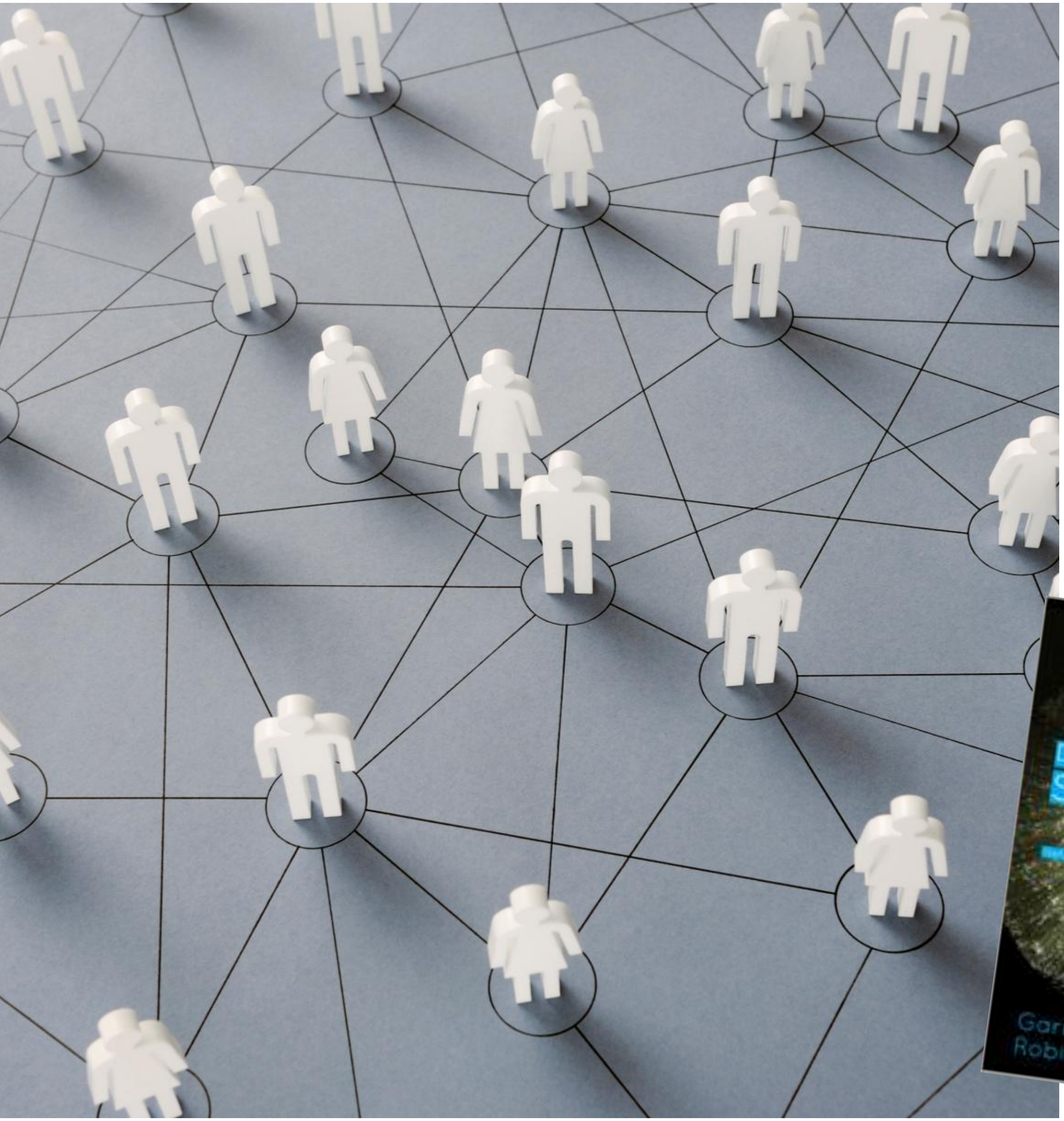
Relational turn →
Novel questions and theory

Novel theoretical considerations

- From endogenous to exogenous explanations
- Interplay of attributional and relational variables
- Going beyond ‘*Hobbessession*’ of assumed independent atomic units

Are relations relevant?

- Contextual
- Certain theoretical frameworks purely relational/attributional
- E.g. post-war development studies:
    - **Modernization school**
      Harrod-Domar domestic capital formation, national policy, stages of economic development
    - **Dependency/world-system tradition**
      External constraints, power-relations, delinking, development of underdevelopment etc



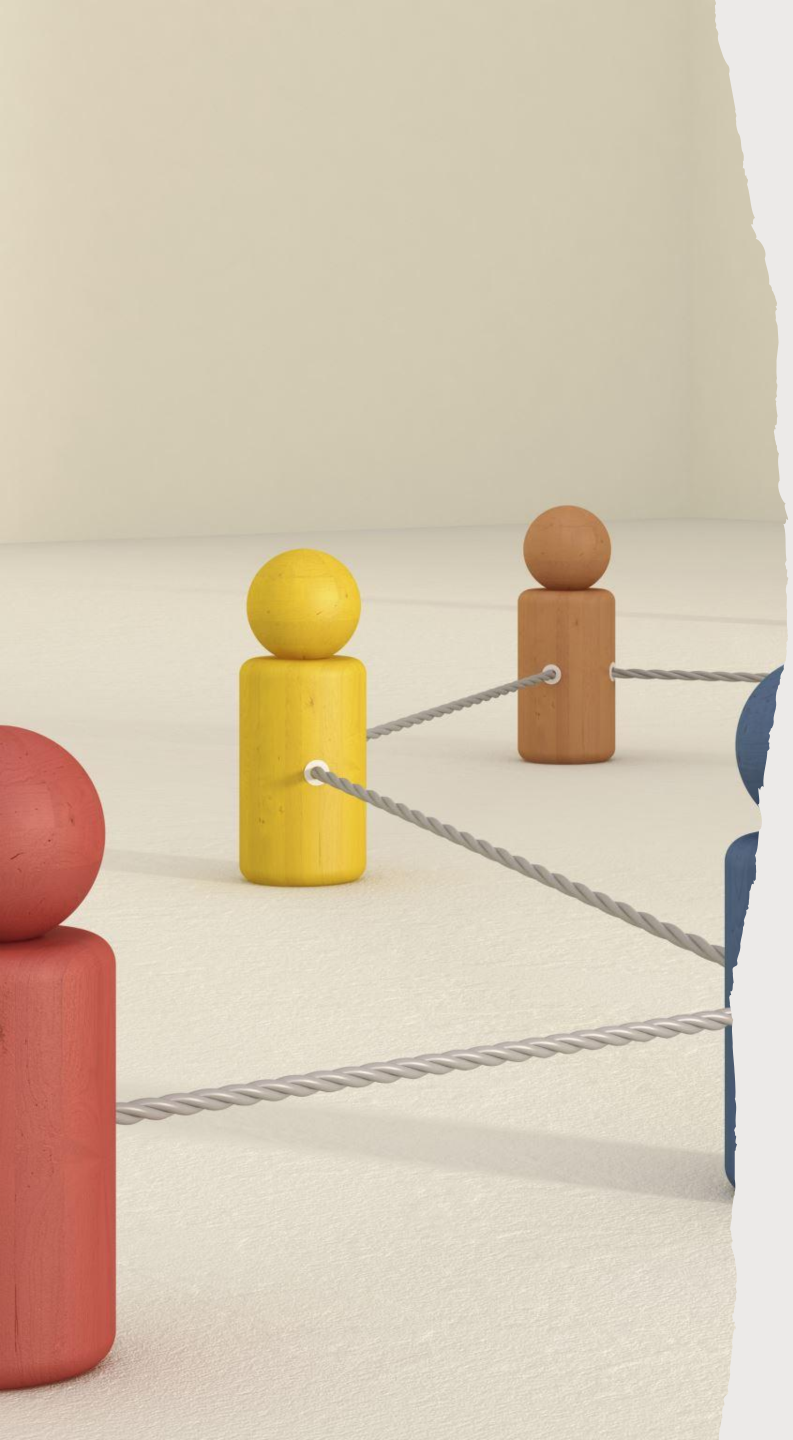
Robins (2015), Doing Social Network Research:
Network-based Research Design for Social Scientists

Reasons for network analysis

- To understand how individuals in certain social position have different individual outcome
- To understand how individuals affect social structure
- To facilitate this: (as always) intricate interplay between:
  - Research question
  - Data
  - Methods
  - Theory

Social groups and communities

- Social group: core notion in sociology and social sciences
  - Where we influence each other
  - Reflecting homophily
  - Development of group norms and standards
- Groups typically defined relationally
  - Could of course also be based on attributes

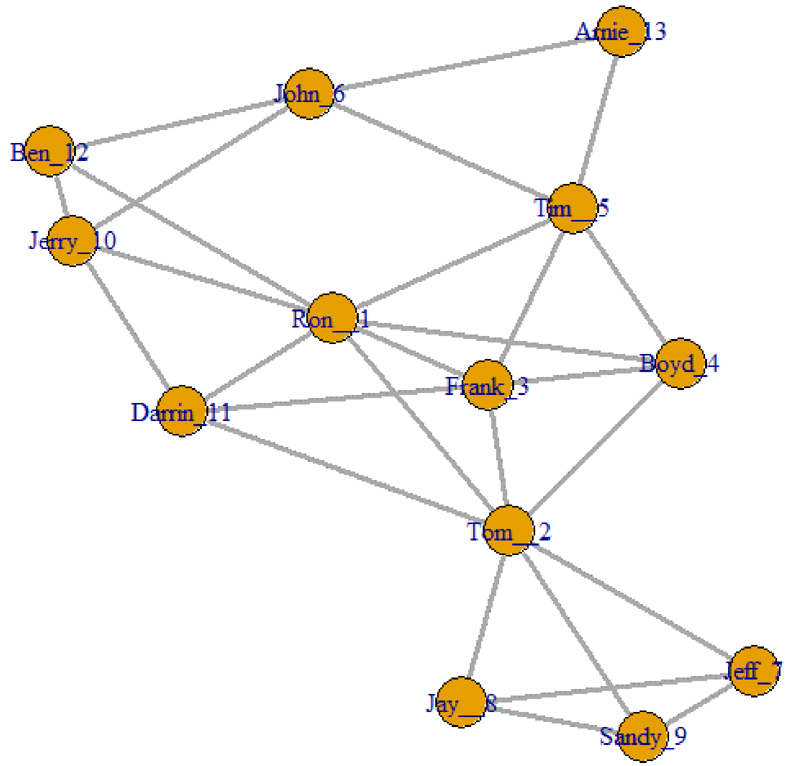


Communities (aka cohesive subgroups)

- The shared relational notion of cohesive subgroup
  - A subset of actors/nodes that are more connected to others within the set than what they are with the outside
- As notion of social group is quite generic
  - Many possible definitions of cohesive subsets
  - A plurality of operationalizations that reflects a plurality of conceptualizations (e.g. Freeman (1992) on the sociological concept of group)
- Let's explore some of these classical types of cohesive subgroups

Meso-level statistics:
Cliques

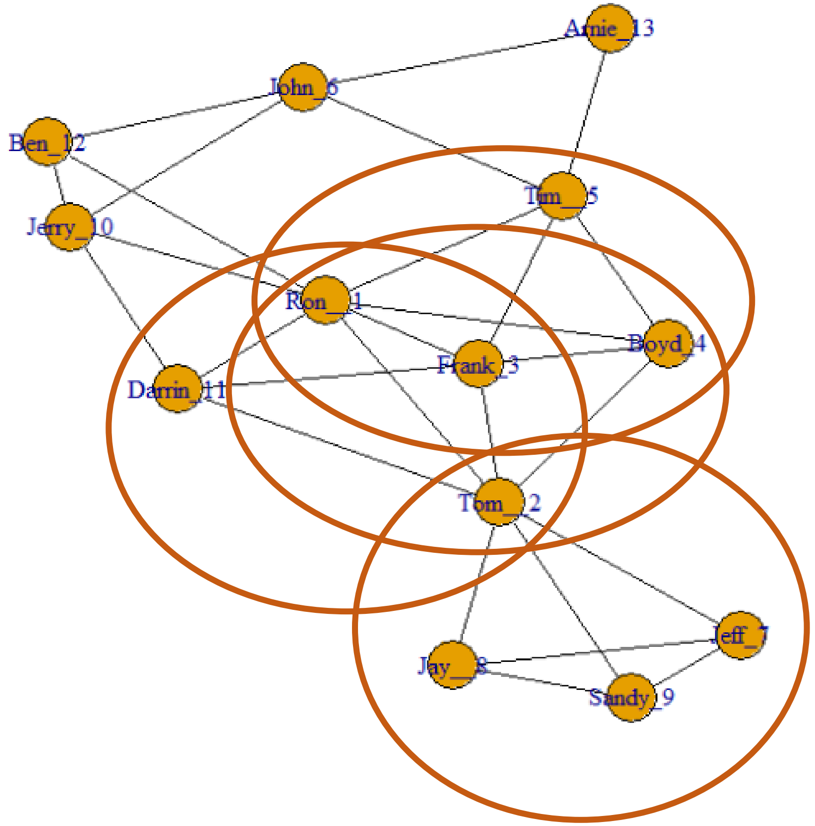
- A maximal complete subgraph of 3 or more actors, where…
  - All actors are connected to each other
  - No other actors that are equally connected to all actors in the clique
- A network typically consists of a lot of cliques of various sizes (particularly triads, i.e. n=3)
- These cliques often overlap each other
- A clique of size 4 contains several cliques of size 3, but the latter ones ignored (see above)



Meso-level statistics:

Cliques: Little League TI

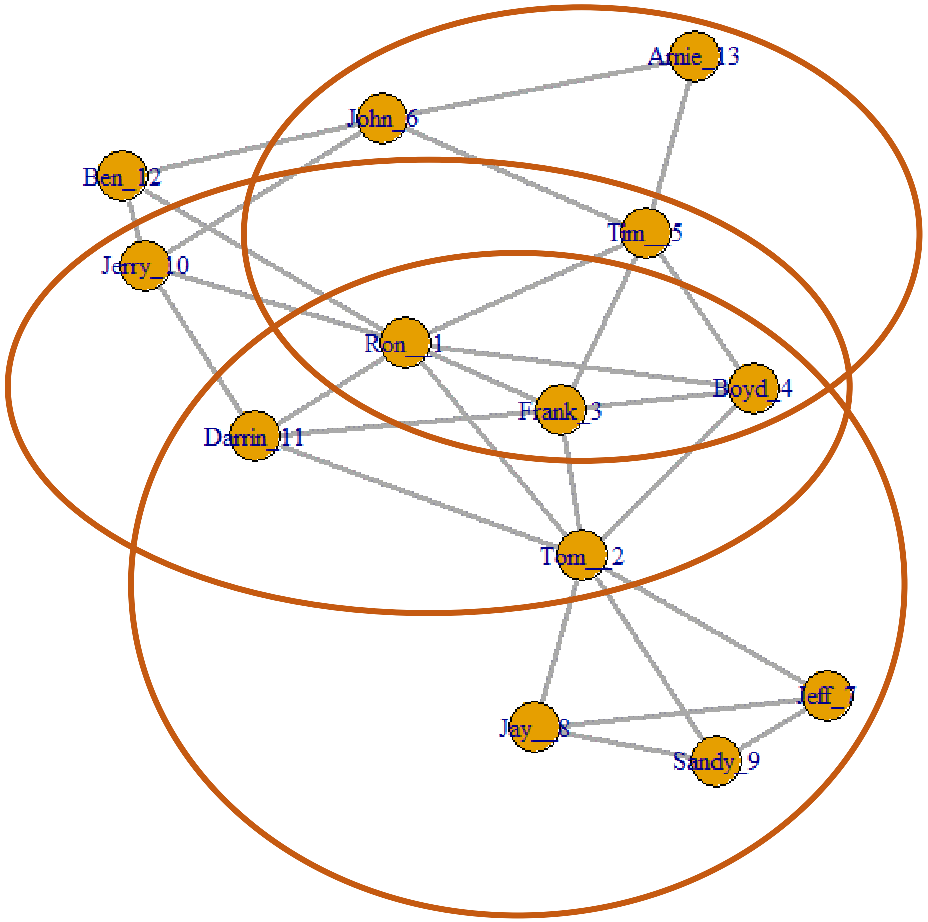
- Igraph function: `max_cliques()`
- 8 cliques
  - 4 cliques of size 4 (marked in red)
    - Tom__2, Ron__1, Frank_3, Boyd_4
    - Tom__2, Ron__1, Frank_3, Darrin_11
    - Frank_3, Ron__1, Tim__5, Boyd_4
    - Tom__2, Sandy_9, Jay__8, Jeff_7
  - 4 cliques of size 3
    - Arnie_13, Tim__5, John_6
    - John_6, Jerry_10, Ben_12
    - Ron__1, Jerry_10, Darrin_11
    - Ron__1, Jerry_10, Ben_12
- Note that there are more triads inside the cliques of size 4, but as they are within larger cliques, they are not maximum cliques



Meso-level statistics:

n-cliques

- A clique requires that each node is maximum one edge away from all others
- The n-clique relaxes this
- An n-clique is a maximum subgraph where distances between all pairs of nodes is no more than n
- Maximum: there can be no other additional node
- When n=1: standard clique
- Varieties:
  - n-clan
    Additional condition: max diameter no longer than n
  - n-club



Example of 2-cliques (there are more)

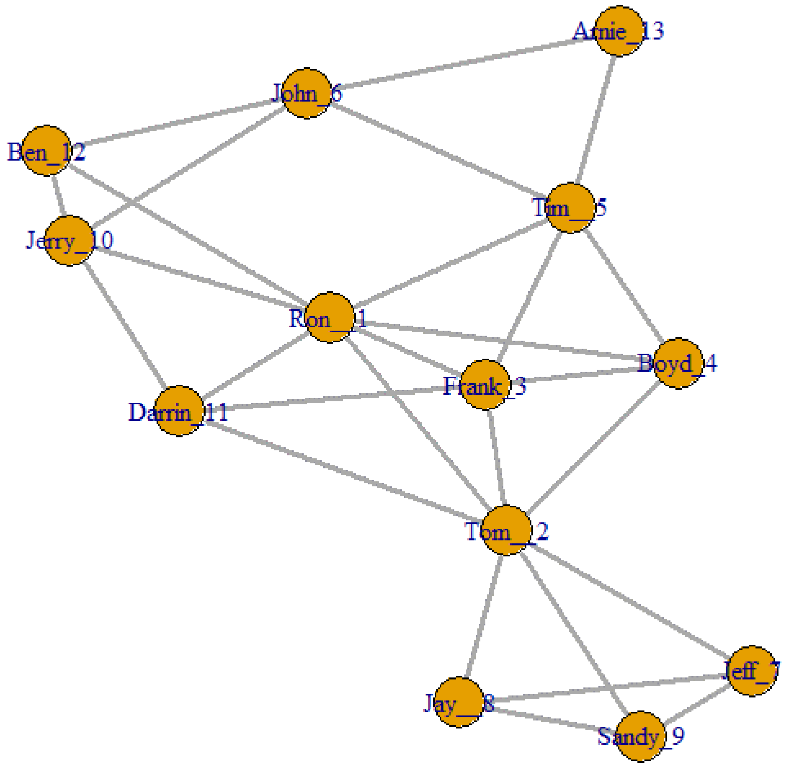
Meso-level statistics:
k-plex

- Maximal subgraph with n nodes in which each node is adjacent to no fewer than (n-k) nodes in subgraph
- Introduced in Seidman & Foster (1978)
- Ignores self-ties
- A clique is a 1-plex
- Testing hypothetical 3-plex (figure to the right)
  - n=8
  - Max 3 missing ties (including self-tie)
  - Must have ties to no fewer than 5 in subgroup
  - ‘can’ only has 4 within-group ties: this is not a 3-plex!

	cw	cysr	jswe	ssr	scw	swra	sw	can	fr	cswj	amh	asw	bjsw	pw	ccq	jgsw	jsp	swg	swhc	ijsw
cw		1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0
cysr	1		0	1	1	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0
jswe	1	0		1	1	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0
ssr	1	1	1		1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	0
scw	1	1	1	1		1	1	1	1	1	0	1	0	0	1	1	0	1	1	0
swra	1	1	1	1	1		1	0	0	0	0	1	0	0	0	0	0	0	0	0
sw	1	1	1	1	1	1		1	1	1	1	1	1	1	0	1	0	1	1	1
can	1	1	0	0	1	0	1		0	0	0	0	0	1	0	0	0	0	0	0
fr	0	1	0	0	1	0	1	0		0	0	0	0	0	0	0	0	0	0	0
cswj	0	0	0	1	1	0	1	0	0		0	0	0	0	0	0	0	0	0	0
amh	0	0	0	0	0	0	1	0	0	0		0	0	0	0	0	0	0	0	0
asw	0	0	1	1	1	1	1	0	0	0	0		0	1	0	0	0	0	0	0
bjsw	0	0	1	0	0	0	1	0	0	0	0	0		0	0	0	0	1	0	0
pw	1	1	0	0	0	0	1	1	0	0	0	1	0		0	0	0	0	0	0
ccq	1	1	0	0	1	0	0	0	0	0	0	0	0	0		0	0	0	0	0
jgsw	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0		0	0	0	0
jsp	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
swg	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0	0		1	0
swhc	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1		0
ijsw	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	

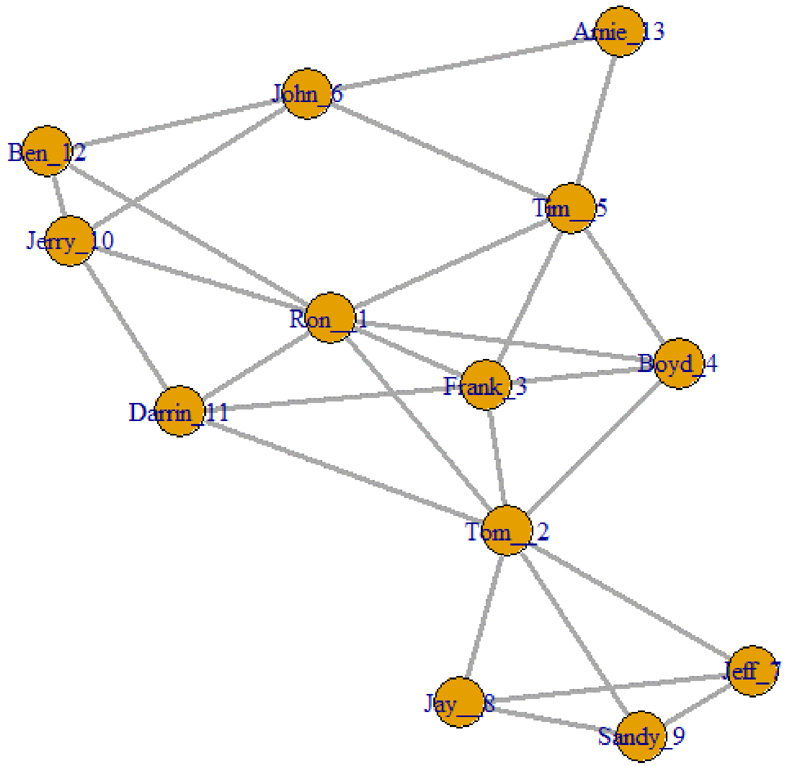
Meso-level statistics:
Communities

- A plethora of SNA-style cohesive subgroup definitions
- From network science: a more practical and generic take on cohesive subgroups
- ‘Community’ instead of ‘cohesive subgroup’
- Less contextual, yet widely used among social scientists as well
- References:
  - Newman & Girwan (2004)
  - Vincent et al (2008)
  - Clauset, Newman & Moore (2004)
  - Coscia et al (2011)
  - Peel et al (2017)
- Most using notion of modularity to measure separation of groups, but different heuristics to get there



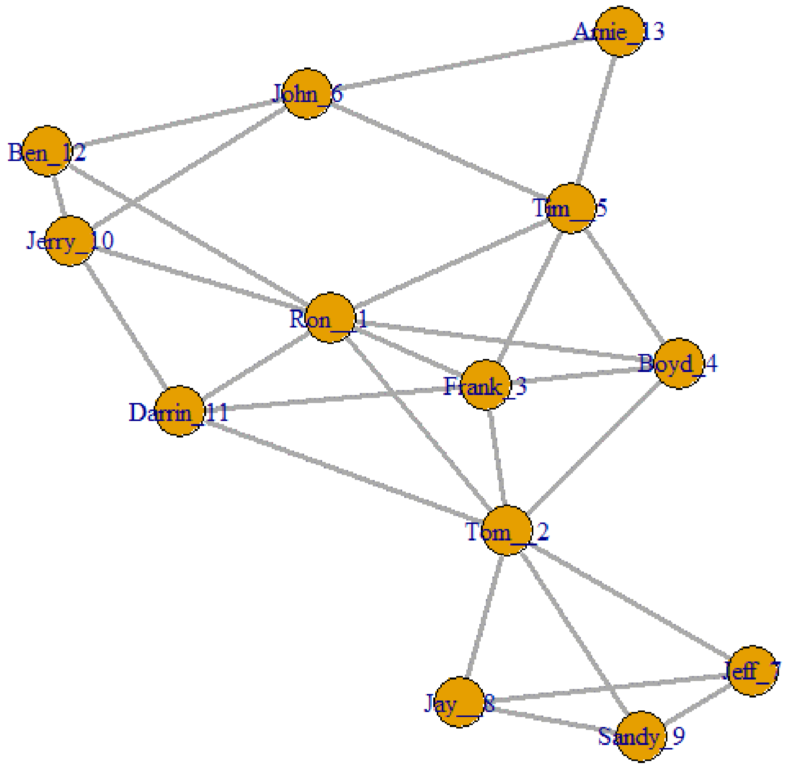
Meso-level statistics:
Newman-Girwan edge betweenness

- Recall betweenness centrality: captures how often node is on shortest path between all pairs of nodes
- Edge betweenness: how often an edge (relation) is on shortest path between all pairs of nodes
- Reasonable that edges with a high edge betweenness are those between communities!
- Newman-Girwan heuristic:
  - Calculate edge betweenness for all edges
  - Remove edges with highest edge betweenness
  - Evaluate how well-separated potential communities are
- Evaluation based on notion of modularity



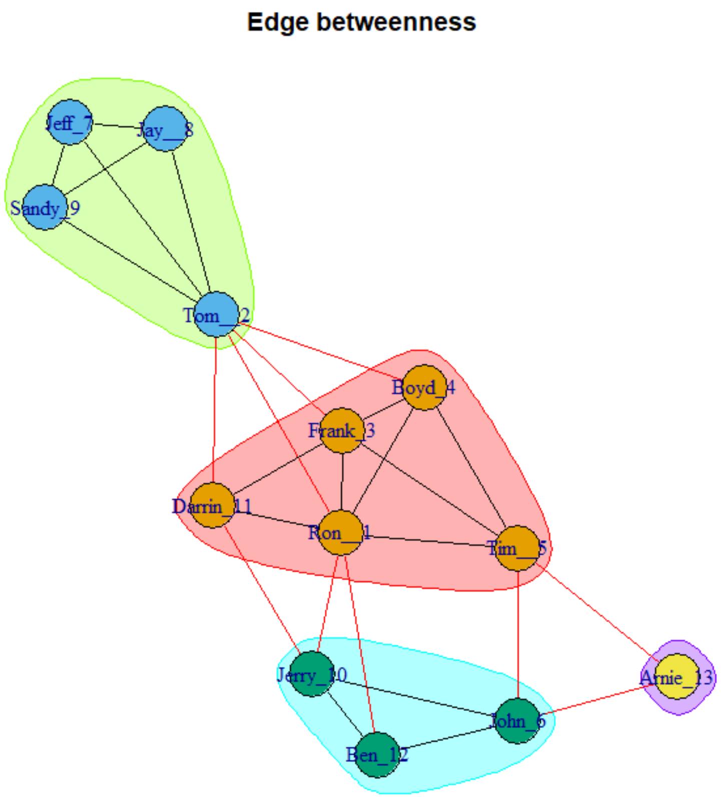
Meso-level statistics:
Modularity

- Measures how well-separated communities are
- Given a partition of nodes into distinct non-overlapping sets
- Modularity reflects the fraction of edges within such sets, minus the expected fraction of edges if they were distributed randomly
- In-build null model consideration
  Clauset, Newman & Moore (2004)



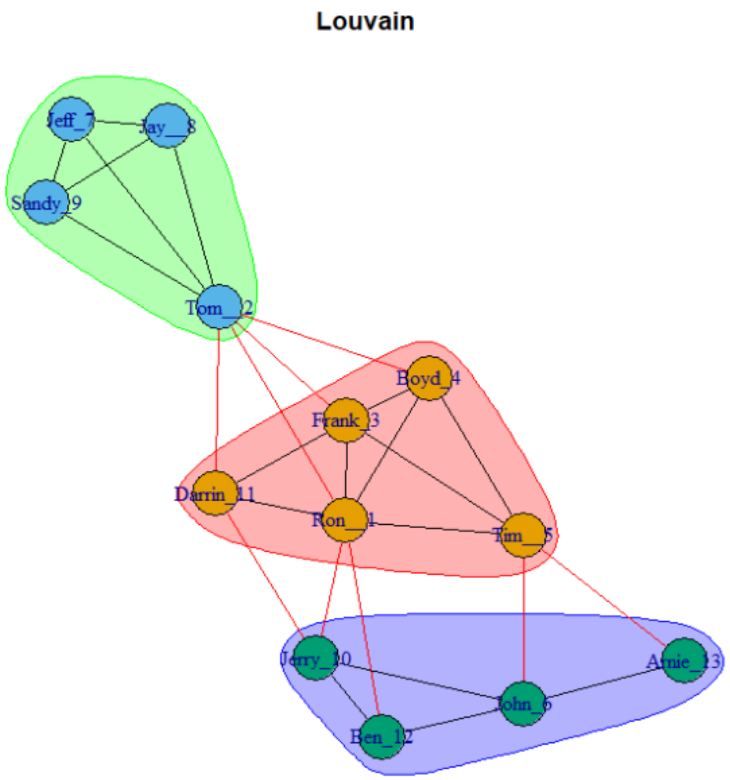
Meso-level statistics:
Newman-Girwan edge betweenness

- Identifying communities using the Newman-Girwan edge betweenness heuristic
- 4 communities identified
- Modularity: 0.285 (not that high)



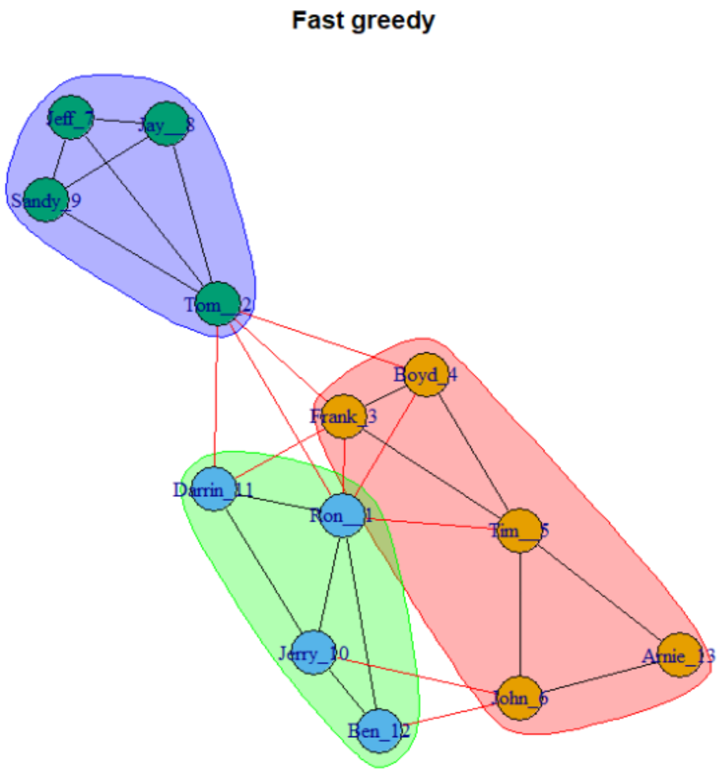
Meso-level statistics:
Louvain heuristic

- Search algorithm to maximize modularity
- Starts with each node as community of its own (singleton)
- Vertices merged when this increases modularity (searching for optimal merge)
- Stops when either single community or no longer possible to increase modularity
- From Blondel, Guillaume, Lambiotte & Lefebvre (2008)
- 3 communities identified
- Modularity: 0.307



Meso-level statistics:
Fast and greedy heuristic

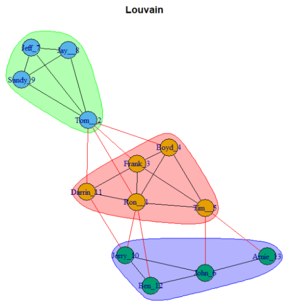
- Fast search algorithm
- Particularly for large networks
- From Clauset, Newman & Moore (2004)
- 3 communities identified
- Modularity: 0.294

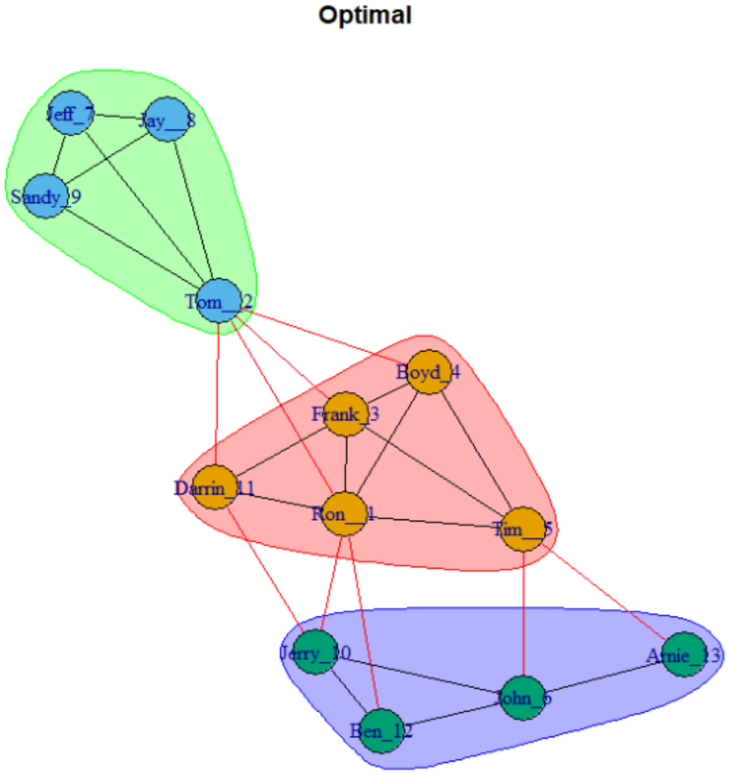


Meso-level statistics:
Optimal modularity

- Naïve search algorithm
- Tests all possible partitions
- Only viable for small networks
- Guaranteed to find partition with highest modularity
- 3 communities identified
- Modularity: 0.307

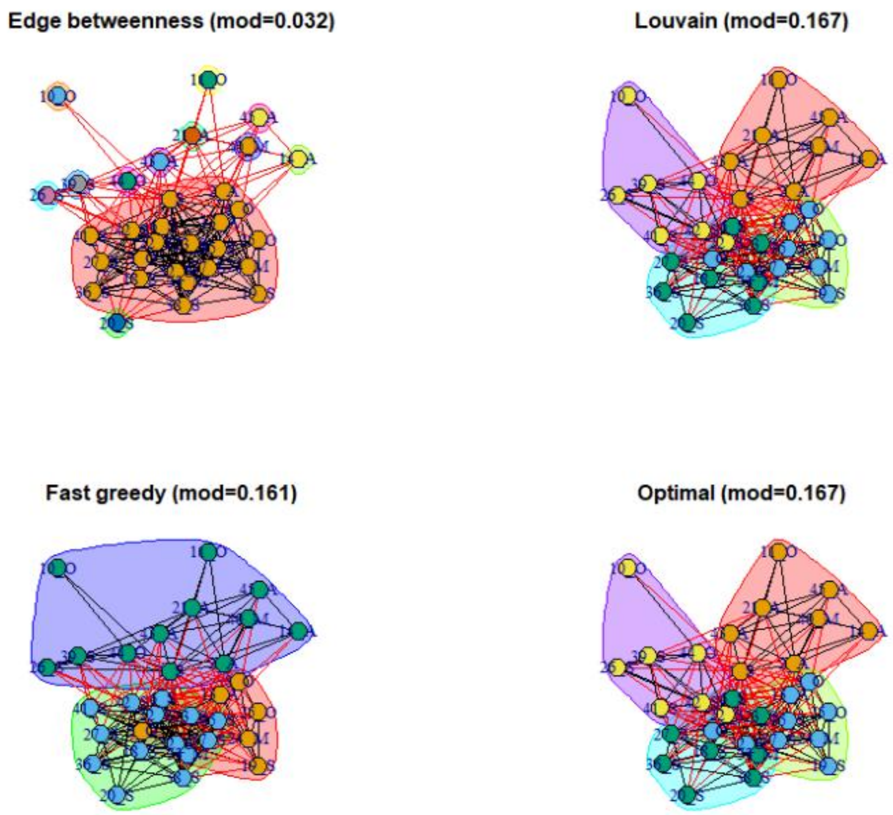
- Identical to what was found with Louvain!





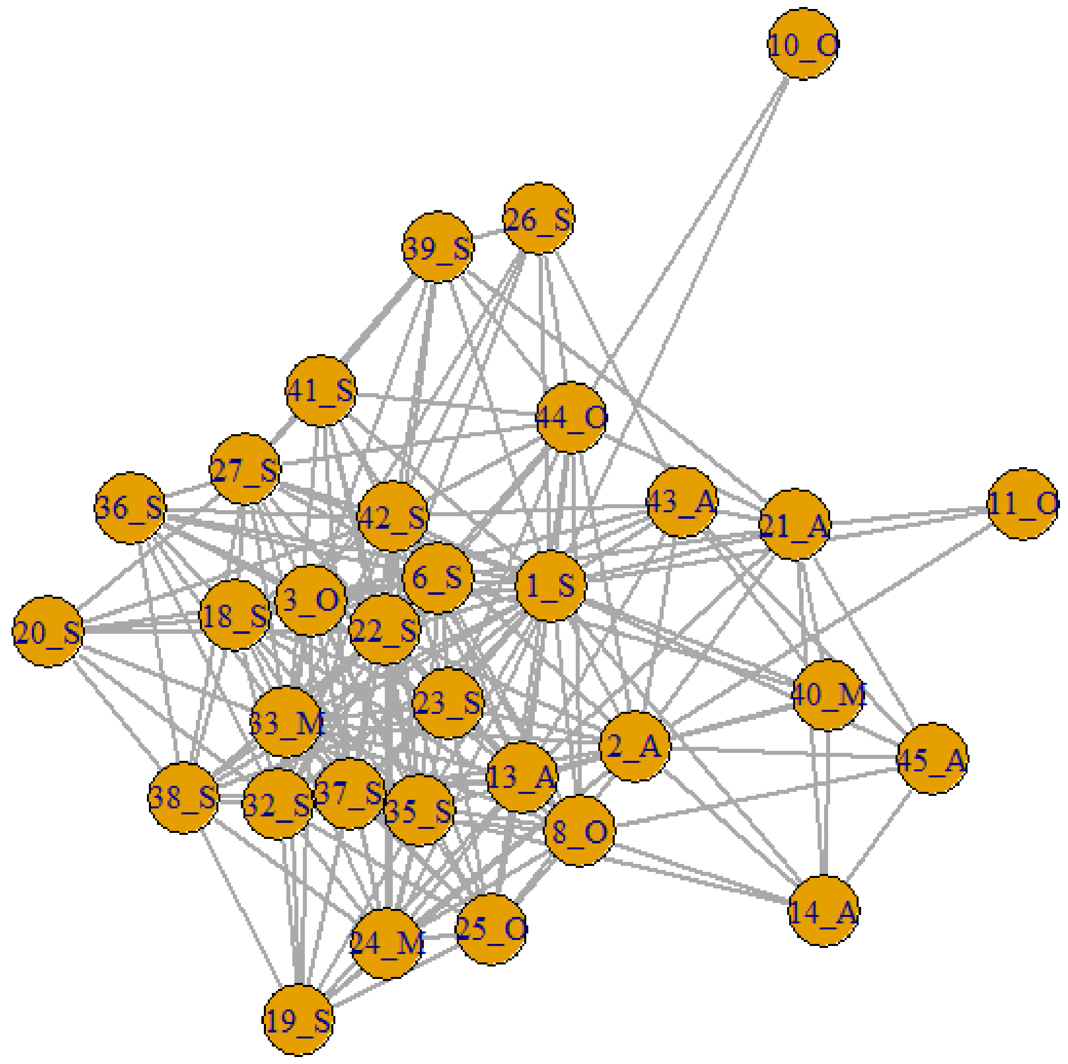
Meso-level statistics:
Communities: interpretational aspects

- So which one to pick?
- Different approaches potentially yields different partitions
- Not always finding most optimal (w.r.t. modularity)
- And is modularity all we need to evaluate community structures in all contexts?
- Peel et al (2017):
  - Seldom a ground truth in community detection
  - No Free Lunch Theorem: “There can be no algorithm that is optimal for all possible community detection tasks”
- Coscia et al (2011): Classification of methods for community detection



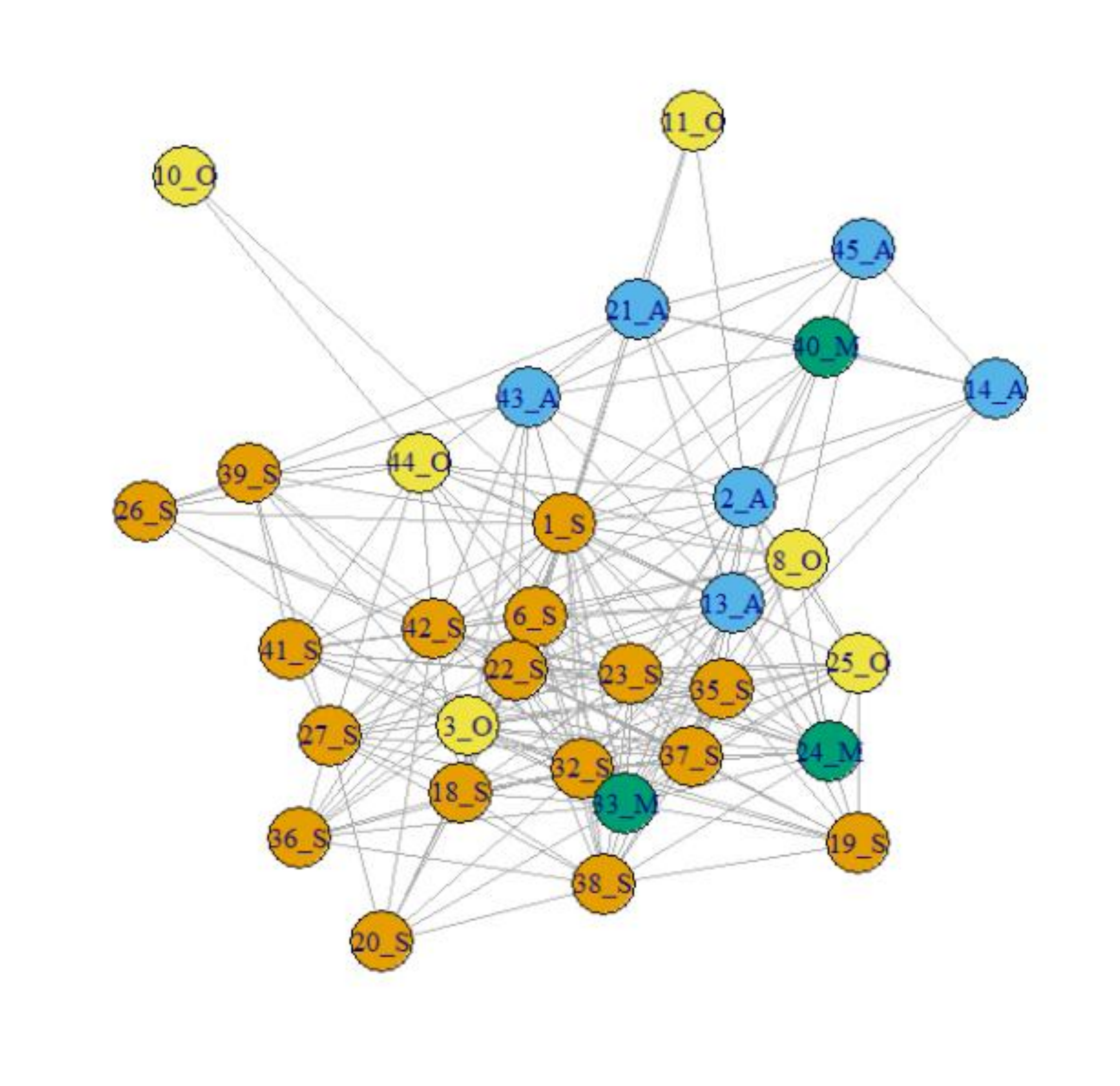
Structures vs. attributes

- Do observed communities reflect nodal attributes?
- Introducing the EIES example data
  - 32 scholars attending an early SNA conference
  - Reported acquaintances (directional, Likert 1-4)
  - Disciplinary backgrounds:
    Sociology (17), Anthropology (6), Mathematics (3), Other (6)
- Dichotomized (>=2) and min-symmetrized
- Pre-conference (also data for post-conference)
- You will explore communities in this network in the lab today!



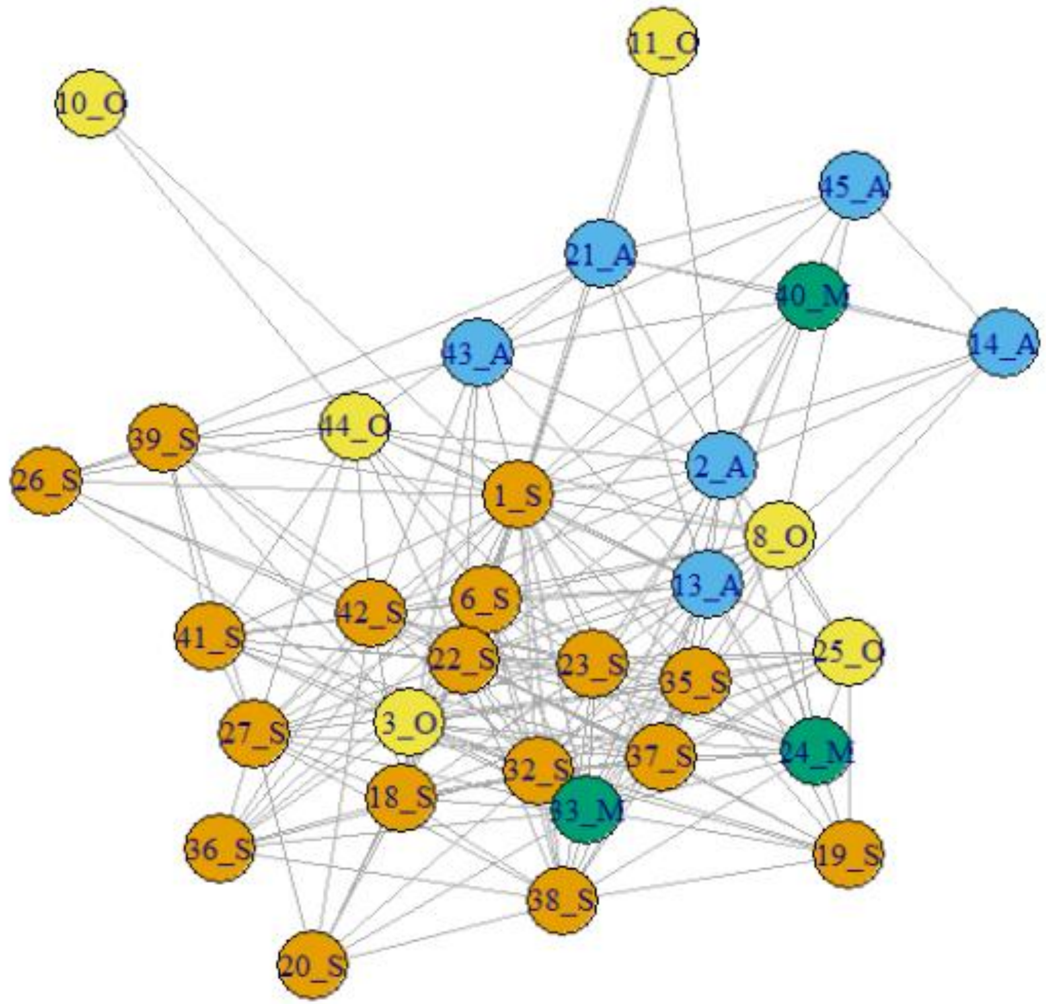
Structures vs. attributes

- Finding some communities in this data
- Do observed communities reflect nodal attributes?
- Ocular inspection:
  - Sociologists indeed seem to stick together
  - Anthropologists as well
  - Maths and Others more dispersed
- But to properly test for assortative mixing, we instead start off with the attributes
- Using assortativity coefficient of Newman (2003) most viable
- High coefficient: connected nodes tend to share same (nominal) attribute



Structures vs. attributes

- Igraph: assortativity_nominal()
- Assortativity coefficient = 0.106
- So slight assortative mixing, but not much!
- In visualization, it indeed looks like a lot of sociologists hanging together
- But bear in mind: many sociologists
  - Quite likely that they would mix by chance as well



Social roles

- Early 1970s, sociologists thought about the notion of social role
- Harrison White: can this be defined in a purely relational form?
- Society IS social structure
  - Multiplex network of inter-personal relations
- Roles and identities in society are “reciprocally defined”
- Only emerging in the interactions

At a hospital

- Four types of actors:
  - Doctors
  - Nurses
  - Admins
  - Patients
- Different roles, as reflected in their interaction patterns:
  - Doctors relate to each other, and to patients
  - Nurses relate to doctors, patients and admins
  - Admins relate to patients
  - Patients relate to all, but not other patients

Conceptual notions of role and position

Schools

- Three types of actors:
  - Kids
  - Teachers
  - Parents
- Different roles, as reflected in their interaction patterns:
  - Parents within a family relate to each other, and to some other parents
  - Parents relate to their own kids, not to other kids
  - Teachers relate to all kids, some parents, some teachers
  - Kids relate to other kids

Conceptual notions of role and position
Patterns of relations

- Seemingly overlap between actor attributes and relational patterns
- Relations not necessarily reflecting cohesive subgroups
  - Patients visiting a hospital have no functional relations with other patients
  - Parents might interact with other parents, but not necessarily
- Being a teacher, parent, doctor etc thus seems to associate with certain relational patterns
  - …that might or might not be cohesive subgroups
- Role analysis and blockmodeling: tools for identifying specific roles on the basis of relational patterns

Conceptual notions of role and position
Patterns of relations

Role-analysis:

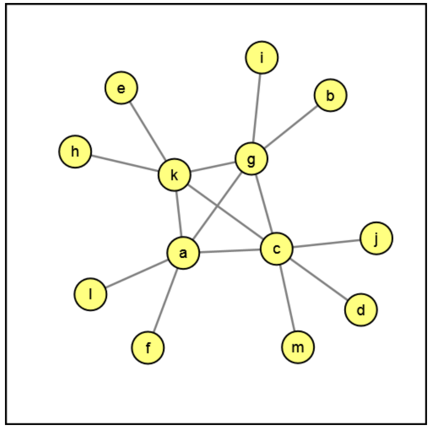
- Identifying sets of nodes that share similar relational patterns
  - Notion of being (structurally) equivalent
- i.e. on the basis of relational patterns alone
- Mapping the overarching patterns of relations between such identified sets of equivalent nodes

Blockmodeling:

- The quintessential technique for role analysis

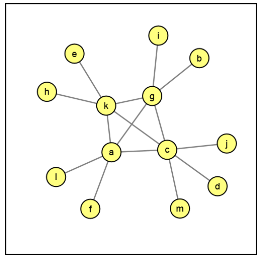
Meso-level statistics: Role-analysis and blockmodeling

- Example: feudal interaction structure (Galtung 1971)
- Ocular inspection:
  - A set of 4 “central” and connected actors in the middle
  - Rest of actors “hanging on”, only connected to these 4 central ones
- Let us hypothesize that these correspond to two different “positions” (subsets of actors):
  - P1 = {a, c, g, k}
  - P2 = {b, d, e, f, h, I, j, l, m}



	a	b	c	d	e	f	g	h	l	j	k	l	m
a		0	1	0	0	1	1	0	0	0	1	1	0
b	0		0	0	0	0	1	0	0	0	0	0	0
c	1	0		1	0	0	1	0	0	1	1	0	1
d	0	0	1		0	0	0	0	0	0	0	0	0
e	0	0	0	0		0	0	0	0	0	1	0	0
f	1	0	0	0	0		0	0	0	0	0	0	0
g	1	1	1	0	0	0		0	1	0	1	0	0
h	0	0	0	0	0	0	0		0	0	1	0	0
l	0	0	0	0	0	0	1	0		0	0	0	0
j	0	0	1	0	0	0	0	0	0		0	0	0
k	1	0	1	0	1	0	1	1	0	0		0	0
l	1	0	0	0	0	0	0	0	0	0	0		0
m	0	0	1	0	0	0	0	0	0	0	0	0	

Meso-level statistics:
Role-analysis and blockmodeling

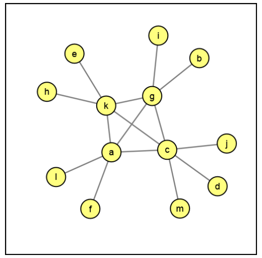


- Let us hypothesize that these correspond to two different “positions” (subsets of actors):
  - P1 = {a, c, g, k}
  - P2 = {b, d, e, f, h, I, j, l, m}
- We sort the sociomatrix according to these two positions
- Highlighting the positions, we arrive at a blockmodel
- We note that a total of four blocks (2^2) emerge:
  - P1 – P1
    This is completely filled with ties (except diagonal)
  - P2 – P2
    There are no ties whatsoever in this block
  - P1 – P2
    This block has one tie on each of its row and columns
  - P2 – P1
    Same as for P1 – P2

	a	b	c	d	e	f	g	h	I	j	k	l	m
a		0	1	0	0	1	1	0	0	0	1	1	0
b	0		0	0	0	0	1	0	0	0	0	0	0
c	1	0		1	0	0	1	0	0	1	1	0	1
d	0	0	1		0	0	0	0	0	0	0	0	0
e	0	0	0	0		0	0	0	0	0	1	0	0
f	1	0	0	0	0		0	0	0	0	0	0	0
g	1	1	1	0	0	0		0	1	0	1	0	0
h	0	0	0	0	0	0	0		0	0	1	0	0
I	0	0	0	0	0	0	1	0		0	0	0	0
j	0	0	1	0	0	0	0	0	0		0	0	0
k	1	0	1	0	1	0	1	1	0	0		0	0
l	1	0	0	0	0	0	0	0	0	0	0		0
m	0	0	1	0	0	0	0	0	0	0	0	0	

		P1				P2								
		a	g	k	c	l	f	I	b	e	h	m	j	d
P1	a		1	1	1	1	1	0	0	0	0	0	0	0
	g	1		1	1	0	0	1	1	0	0	0	0	0
	k	1	1		1	0	0	0	0	1	1	0	0	0
	c	1	1	1		0	0	0	0	0	0	1	1	1
P2	l	1	0	0	0		0	0	0	0	0	0	0	0
	f	1	0	0	0	0		0	0	0	0	0	0	0
	I	0	1	0	0	0	0		0	0	0	0	0	0
	b	0	1	0	0	0	0	0		0	0	0	0	0
	e	0	0	1	0	0	0	0	0		0	0	0	0
	h	0	0	1	0	0	0	0	0	0		0	0	0
	m	0	0	0	1	0	0	0	0	0	0		0	0
	j	0	0	0	1	0	0	0	0	0	0	0		0
	d	0	0	0	1	0	0	0	0	0	0	0	0	

Meso-level statistics:
Role-analysis and blockmodeling



	a	b	c	d	e	f	g	h	i	j	k	l	m
a		0	1	0	0	1	1	0	0	0	1	1	0
b	0		0	0	0	0	1	0	0	0	0	0	0
c	1	0		1	0	0	1	0	0	1	1	0	1
d	0	0	1		0	0	0	0	0	0	0	0	0
e	0	0	0	0		0	0	0	0	0	1	0	0
f	1	0	0	0	0		0	0	0	0	0	0	0
g	1	1	1	0	0	0		0	1	0	1	0	0
h	0	0	0	0	0	0	0		0	0	1	0	0
i	0	0	0	0	0	0	1	0		0	0	0	0
j	0	0	1	0	0	0	0	0	0		0	0	0
k	1	0	1	0	1	0	1	1	0	0		0	0
l	1	0	0	0	0	0	0	0	0	0	0		0
m	0	0	1	0	0	0	0	0	0	0	0	0	

		P1				P2								
		a	g	k	c	i	f	l	b	e	h	m	j	d
P1	a		1	1	1	1	1	0	0	0	0	0	0	0
	g	1		1	1	0	0	1	1	0	0	0	0	0
	k	1	1		1	0	0	0	0	1	1	0	0	0
	c	1	1	1		0	0	0	0	0	0	1	1	1
P2	i	1	0	0	0		0	0	0	0	0	0	0	0
	f	1	0	0	0	0		0	0	0	0	0	0	0
	l	0	1	0	0	0	0		0	0	0	0	0	0
	b	0	1	0	0	0	0	0		0	0	0	0	0
	e	0	0	1	0	0	0	0	0		0	0	0	0
	h	0	0	1	0	0	0	0	0	0		0	0	0
	m	0	0	0	1	0	0	0	0	0	0		0	0
	j	0	0	0	1	0	0	0	0	0	0	0		0
	d	0	0	0	1	0	0	0	0	0	0	0	0	

- The four blocks that emerge in this hypothetical blockmodel represent three ideal blocks in blockmodeling:

Structural equivalence blockmodeling

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

	1	1	1
1		1	1
1	1		1
1	1	1	

The complete block (com)

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

	0	0	0
0		0	0
0	0		0
0	0	0	

The null block (nul)

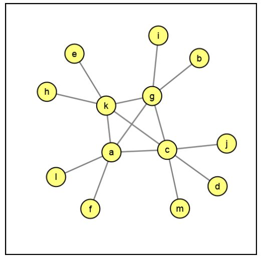
1	0	1	0	0
0	0	0	1	0
0	1	0	0	1
0	0	1	0	0

	1	1	0
1		1	0
0	0		1
0	1	0	

The regular block (reg)

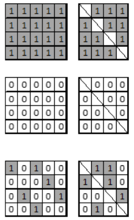
Regular equivalence blockmodeling

Meso-level statistics:
Role-analysis and blockmodeling



- As each of these blocks perfectly match an ideal block type, we can 'collapse' the blockmodel into a so-called blockimage – or an image graph
- Nodes in the image graph now represent positions (i.e. subsets of equivalent actors)
- The blockimage and image graph thus captures the underlying anatomy of the network
- In this case, this corresponds to a core-periphery network
- Roles:
    - Role of P1 is to be connected to others in P1, and having sporadic ties to and from P2
    - Role of P2 is to have no ties to others in P2, and singular ties to P1

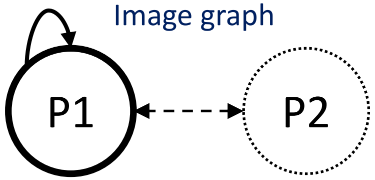
		P1				P2								
		a	g	k	c	l	f	l	b	e	h	m	j	d
P1	a		1	1	1	1	1	0	0	0	0	0	0	0
	g	1		1	1	0	0	1	1	0	0	0	0	0
	k	1	1		1	0	0	0	0	1	1	0	0	0
	c	1	1	1		0	0	0	0	0	0	1	1	1
P2	l	1	0	0	0		0	0	0	0	0	0	0	0
	f	1	0	0	0	0		0	0	0	0	0	0	0
	l	0	1	0	0	0	0		0	0	0	0	0	0
	b	0	1	0	0	0	0	0		0	0	0	0	0
	e	0	0	1	0	0	0	0	0		0	0	0	0
	h	0	0	1	0	0	0	0	0	0		0	0	0
	m	0	0	0	1	0	0	0	0	0	0		0	0
	j	0	0	0	1	0	0	0	0	0	0	0		0
	d	0	0	0	1	0	0	0	0	0	0	0	0	



The complete block (com)

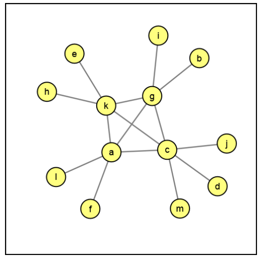
The null block (nul)

The regular block (reg)



	P1	P2
P1	com	reg
P2	reg	nul

Meso-level statistics:
Role-analysis and blockmodeling



Why blockmodeling and role-analysis?

- A blockmodel reduces a complex network into its fundamental structure (functional anatomy)
- In the Galtung network, we (obviously) find an ideal core-periphery structure
- For a network many times the size, the same block image would appear
- Allows for comparing complex networks that might differ in size, context, distributions

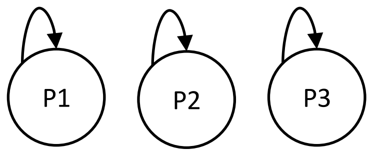
		P1				P2								
		a	g	k	c	l	f	l	b	e	h	m	j	d
P1	a		1	1	1	1	1	0	0	0	0	0	0	0
	g	1		1	1	0	0	1	1	0	0	0	0	0
	k	1	1		1	0	0	0	0	1	1	0	0	0
	c	1	1	1		0	0	0	0	0	0	1	1	1
P2	l	1	0	0	0		0	0	0	0	0	0	0	0
	f	1	0	0	0	0		0	0	0	0	0	0	0
	l	0	1	0	0	0	0		0	0	0	0	0	0
	b	0	1	0	0	0	0	0		0	0	0	0	0
	e	0	0	1	0	0	0	0	0		0	0	0	0
	h	0	0	1	0	0	0	0	0	0		0	0	0
	m	0	0	0	1	0	0	0	0	0	0		0	0
	j	0	0	0	1	0	0	0	0	0	0	0		0
	d	0	0	0	1	0	0	0	0	0	0	0	0	

	P1	P2
P1	com	reg
P2	reg	nul

Meso-level statistics:

Role-analysis and blockmodeling

- Galtung example turned out to be a core-periphery structure
- There are a multitude of other possibilities
  - Hierarchies
  - Transitive structures
  - Cohesive subgroups
  - Something very specific
- An ideal community structure is a special case (one possible outcome) of blockmodeling
- So: NOT EXPLICITLY looking for community structure, OR core-periphery structure
  - Rather: look for regularities in relational patterns
  - …which COULD be a community structure, a core-periphery structure, or something different



	P1	**P2**	**P3**
P1	com	nul	nul
P2	nul	com	nul
P3	nul	nul	com

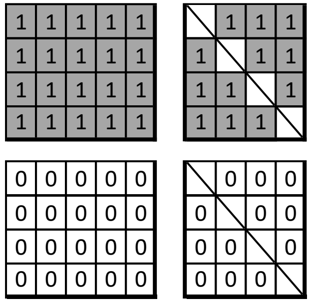
Meso-level statistics:
Types of equivalences

What is meant by equivalence in a network?

Different notions and usages exist

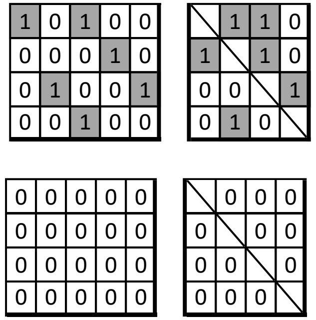
Structural equivalence

- Two actors are structurally equivalent if they have identical sets of ties to the other actors
- Ideal blocks: complete and null blocks



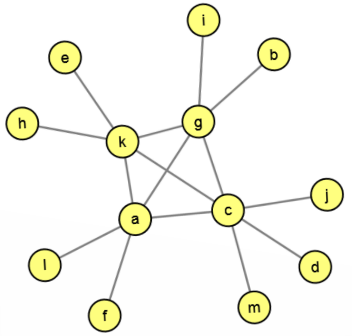
Regular equivalence

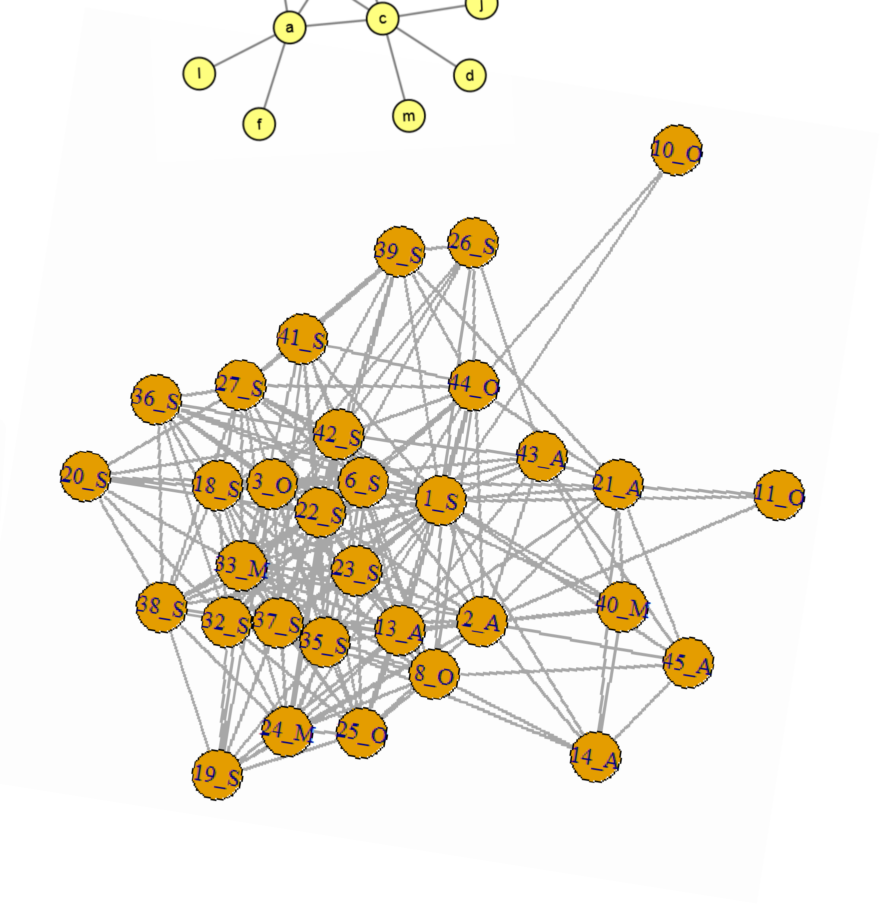
- Two actors are regular equivalent if they have ties to actors that in turn are also regular equivalent (!)
- Ideal blocks: null and regular blocks

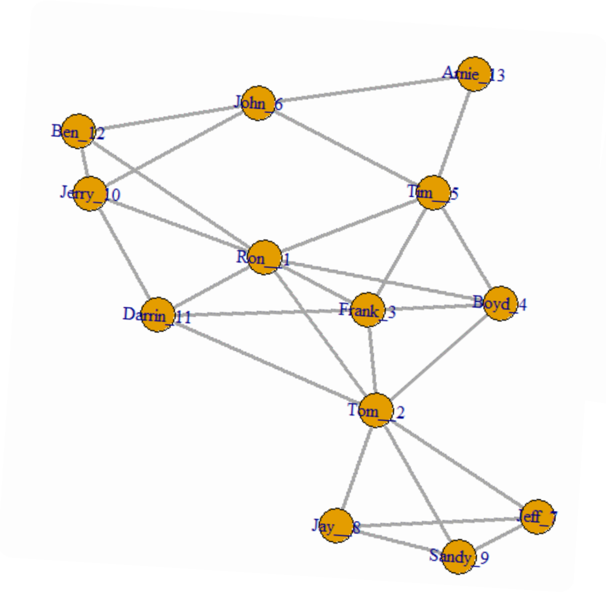


Meso-level statistics:
Types of equivalences

- In the Galtung example, we managed to perfectly fit the network to ideal blocks – and on first attempt!
- Real-world networks are more messy
- Even our small example networks can be messy!
- The ambition with blockmodeling: find the best possible fit
- Two broad approaches for partitioning a network into equivalent positions
  - Direct methods
  - Indirect methods
  - (And also hypothesis testing)

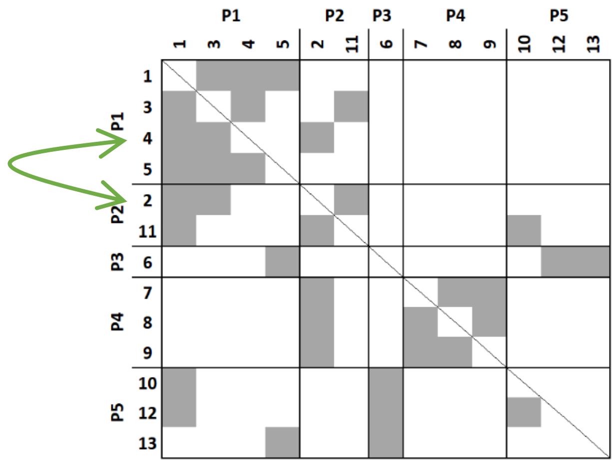


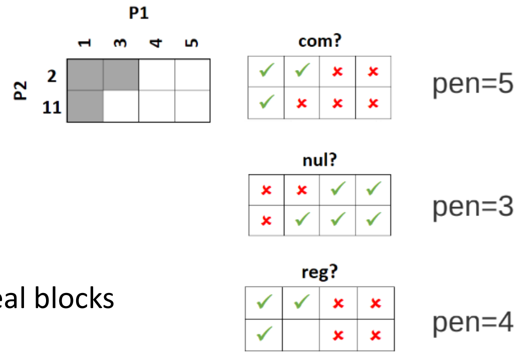




Meso-level statistics:
Direct blockmodeling

- Starts with a random starting partition for a specified number of positions
- Local optimization search algorithm to move/swap actors between positions
- Using goodness-of-fit (criteria) function to evaluate resulting blockmodels
- Actual block image can be pre-specified (e.g. if explicitly searching for a core-periphery structure)
- Can also be flexible: then testing all available ideal blocks
- Typical goodness-of-fit measures
  - Hamming distances (number of inconsistencies)
  - (Weighted) correlation coefficients (also for valued networks)





Advantage:
Deemed better
Allows for capturing complicated ideal blocks (generalized blockmodeling)

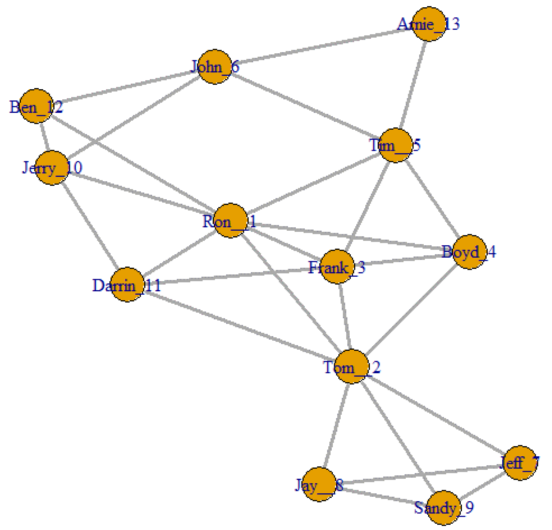
Disadvantage:
Computationally heavy
Potentially finding local minima

Meso-level statistics:
Indirect blockmodeling

- Starts by calculating/determining a suitable indirect measure of equivalence
- Most often structural equivalence
- This results in a new (dis)similarity matrices indicating how structurally equivalent each pair of nodes is
- Apply hierarchical clustering
- Choose suitable number of positions
- Create and interpret blockmodel

Advantage:
Works for large networks
Analytical, not computational

Disadvantage:
Restricted to certain notions of equivalence
Generalized blockmodeling: nope!
Lots of choices in the analysis



	Ron__1	**Tom__2**	**Frank_3**	**Boyd_4**	**Tim__5**	**John_6**	**Jeff_7**	**Jay__8**	**Sandy_9**	**Jerry_10**	**Darrin_11**	**Ben_12**	**Arnie_13**
Ron__1		0	1	1	1	0	0	0	0	0	0	0	0
Tom__2	1		1	0	0	0	0	0	0	0	1	0	0
Frank_3	1	0		1	0	0	0	0	0	0	1	0	0
Boyd_4	1	1	1		0	0	0	0	0	0	0	0	0
Tim__5	1	0	1	1		0	0	0	0	0	0	0	0
John_6	0	0	0	0	1		0	0	0	0	0	1	1
Jeff_7	0	1	0	0	0	0		1	1	0	0	0	0
Jay__8	0	1	0	0	0	0	1		1	0	0	0	0
Sandy_9	0	1	0	0	0	0	1	1		0	0	0	0
Jerry_10	1	0	0	0	0	1	0	0	0		0	0	0
Darrin_11	1	1	0	0	0	0	0	0	0	1		0	0
Ben_12	1	0	0	0	0	1	0	0	0	1	0		0
Arnie_13	0	0	0	0	1	1	0	0	0	0	0	0	

Meso-level statistics:
Indirect structural equivalence

Structural equivalence

- That two actors have the same relational patterns to the other actors
- Let's pick two actors – Tom_2 and Frank_3 – and look at their respective rows and columns:

Rows:

Tom__2	1		1	0	0	0	0	0	0	0	1	0	0
	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Frank_3	1	0		1	0	0	0	0	0	0	1	0	0

Columns:

Tom__2	0		0	1	0	0	1	1	1	0	1	0	0
	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓	
Frank_3	1	1		1	1	0	0	0	0	0	0	0	0

✓ 14

✗ 7 (+1)

- Ignore self-ties
- Check for reciprocity (once)

	Ron__1	Tom__2	Frank_3	Boyd_4	Tim__5	John_6	Jeff_7	Jay__8	Sandy_9	Jerry_10	Darrin_11	Ben_12	Arnie_13
Ron__1		0	1	1	1	0	0	0	0	0	0	0	0
Tom__2	1		1	0	0	0	0	0	0	0	1	0	0
Frank_3	1	0		1	0	0	0	0	0	0	1	0	0
Boyd_4	1	1	1		0	0	0	0	0	0	0	0	0
Tim__5	1	0	1	1		0	0	0	0	0	0	0	0
John_6	0	0	0	0	1		0	0	0	0	0	1	1
Jeff_7	0	1	0	0	0	0		1	1	0	0	0	0
Jay__8	0	1	0	0	0	0	1		1	0	0	0	0
Sandy_9	0	1	0	0	0	0	1	1		0	0	0	0
Jerry_10	1	0	0	0	0	1	0	0	0		0	0	0
Darrin_11	1	1	0	0	0	0	0	0	0	1		0	0
Ben_12	1	0	0	0	0	1	0	0	0	1	0		0
Arnie_13	0	0	0	0	1	1	0	0	0	0	0	0	

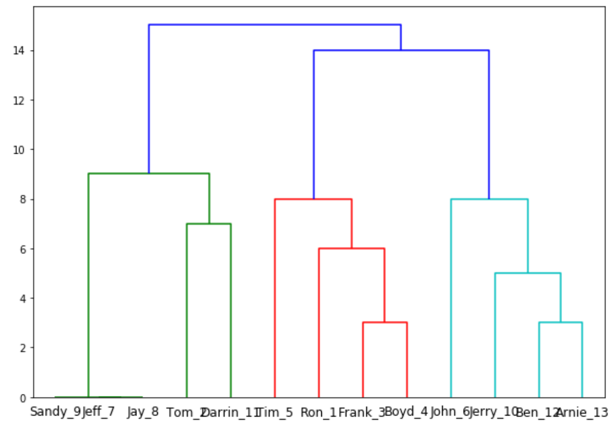
- Structural equivalence dissimilarity for Tom_2 and Frank_3 is 8
- Repeat for all pairs to create dissimilarity matrix

Meso-level statistics:
Indirect structural equivalence

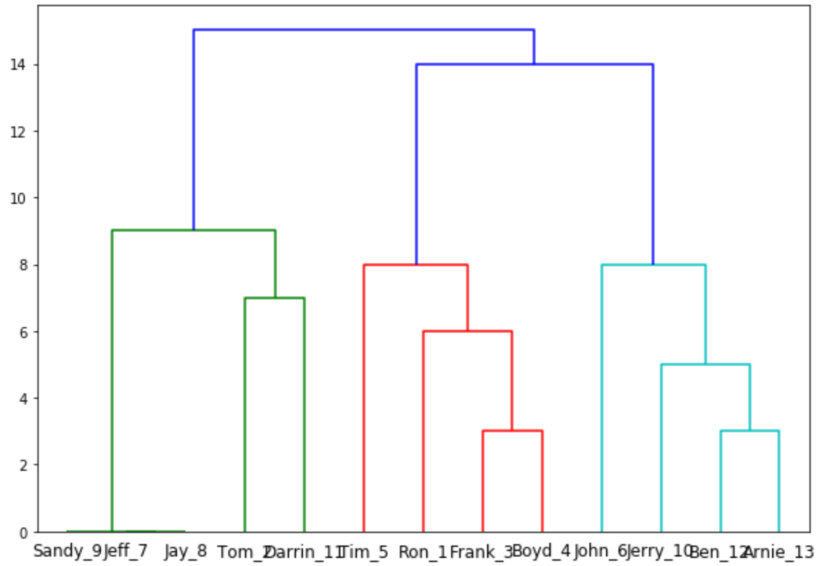
- Start off with actual network
- Calculate indirect measure of structural equivalence
  - Here, using Hamming distance dissimilarity (number of inconsistencies)
- Apply clustering algorithm to identify structurally equivalent actors
  - Here, using (unweighted) average-link clustering

	Ron__1	Tom__2	Frank_3	Boyd_4	Tim__5	John_6	Jeff_7	Jay__8	Sandy_9	Jerry_10	Darrin_11	Ben_12	Arnie_13
Ron__1		0	1	1	1	0	0	0	0	0	0	0	0
Tom__2	1		1	0	0	0	0	0	0	0	1	0	0
Frank_3	1	0		1	0	0	0	0	0	0	1	0	0
Boyd_4	1	1	1		0	0	0	0	0	0	0	0	0
Tim__5	1	0	1	1		0	0	0	0	0	0	0	0
John_6	0	0	0	0	1		0	0	0	0	0	1	1
Jeff_7	0	1	0	0	0	0		1	1	0	0	0	0
Jay__8	0	1	0	0	0	0	1		1	0	0	0	0
Sandy_9	0	1	0	0	0	0	1	1		0	0	0	0
Jerry_10	1	0	0	0	0	1	0	0	0		0	0	0
Darrin_11	1	1	0	0	0	0	0	0	0	1		0	0
Ben_12	1	0	0	0	0	1	0	0	0	1	0		0
Arnie_13	0	0	0	0	1	1	0	0	0	0	0	0	

	Ron__1	Tom__2	Frank_3	Boyd_4	Tim__5	John_6	Jeff_7	Jay__8	Sandy_9	Jerry_10	Darrin_11	Ben_12	Arnie_13
Ron__1		11	5	6	8	10	15	15	15	9	10	13	11
Tom__2	11		8	9	10	14	8	8	8	8	7	10	11
Frank_3	5	8		3	6	13	12	12	12	9	7	9	10
Boyd_4	6	9	3		5	12	9	9	9	8	5	8	9
Tim__5	8	10	6	5		9	11	11	11	8	9	6	6
John_6	10	14	13	12	9		11	11	11	7	11	6	3
Jeff_7	15	8	12	9	11	11		0	0	9	8	9	8
Jay__8	15	8	12	9	11	11	0		0	9	8	9	8
Sandy_9	15	8	12	9	11	11	0	0		9	8	9	8
Jerry_10	9	8	9	8	8	7	9	9	9		6	3	5
Darrin_11	10	7	7	5	9	11	8	8	8	6		5	8
Ben_12	13	10	9	8	6	6	9	9	9	3	5		3
Arnie_13	11	11	10	9	6	3	8	8	8	5	8	3	



Meso-level statistics:
Indirect structural equivalence



- Examine dendrogram of hierarchical clustering (complete-link here)
- Identify suitable partition
  - P1:
    Tom_2, Jeff_7, Jay_8, Sandy_9, Darrin_11
  - P2:
    Ron_1, Frank_3, Boyd_4, Tim_5
  - P3:
    John_6, Jerry_10, Ben_12, Arnie_13,

Meso-level statistics:
Indirect structural equivalence

- Back to original matrix
- Sort according to identified partitions
- For binary blockmodels, calculating block densities can be useful
- Ideally, block densities should be close to 0 (null) or 1 (complete)

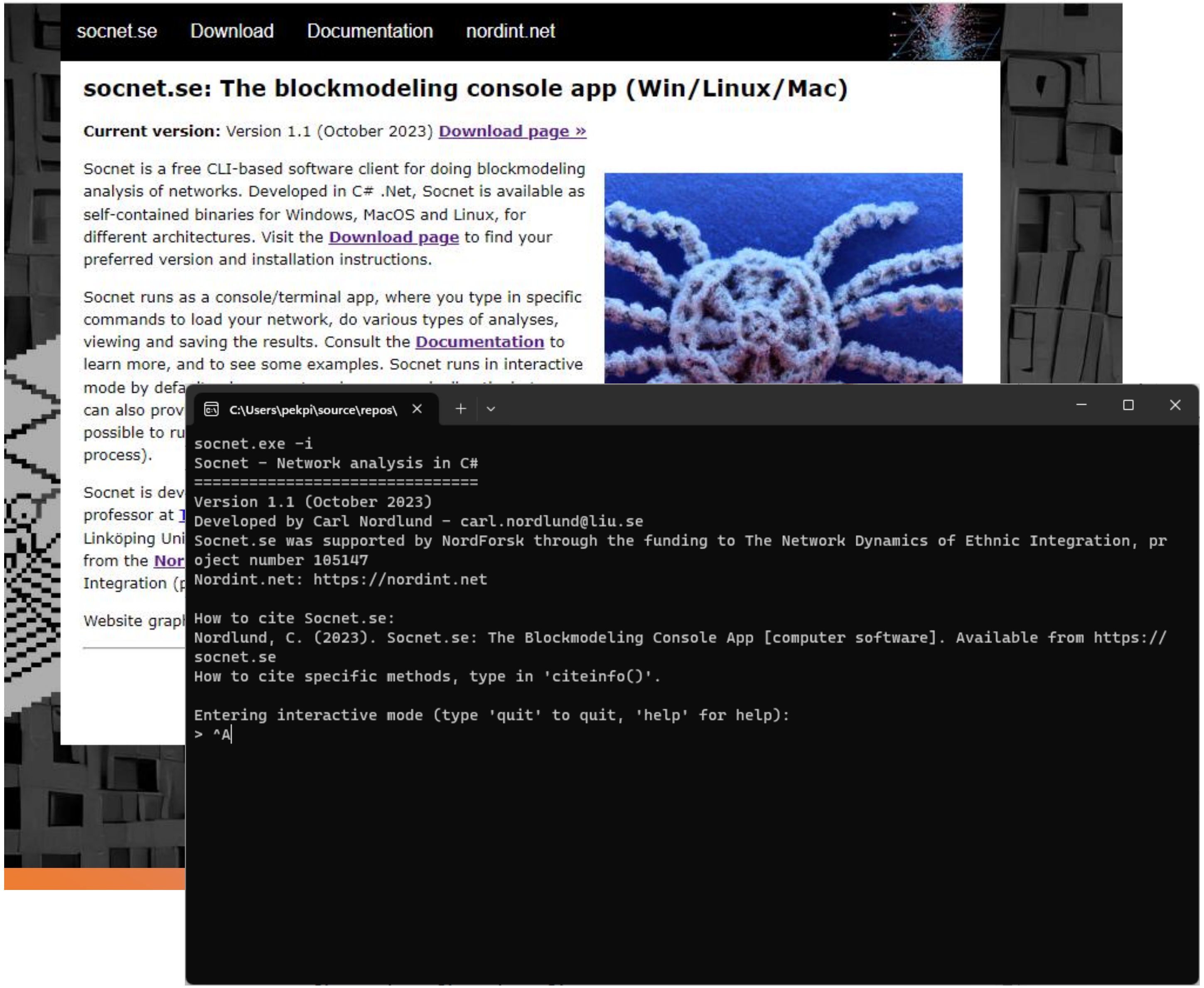
- Not very “crisp” densities
- Community structure (com on diagonal)
- Inconsistencies: 27 (quite high)
  - Which explains why we have uncertain densities

	Ron__1	Tom__2	Frank_3	Boyd_4	Tim__5	John_6	Jeff_7	Jay__8	Sandy_9	Jerry_10	Darrin_11	Ben_12	Arnie_13
Ron__1		0	1	1	1	0	0	0	0	0	0	0	0
Tom__2	1		1	0	0	0	0	0	0	0	1	0	0
Frank_3	1	0		1	0	0	0	0	0	0	1	0	0
Boyd_4	1	1	1		0	0	0	0	0	0	0	0	0
Tim__5	1	0	1	1		0	0	0	0	0	0	0	0
John_6	0	0	0	0	1		0	0	0	0	0	1	1
Jeff_7	0	1	0	0	0	0		1	1	0	0	0	0
Jay__8	0	1	0	0	0	0	1		1	0	0	0	0
Sandy_9	0	1	0	0	0	0	1	1		0	0	0	0
Jerry_10	1	0	0	0	0	1	0	0	0		0	0	0
Darrin_11	1	1	0	0	0	0	0	0	0	1		0	0
Ben_12	1	0	0	0	0	1	0	0	0	1	0		0
Arnie_13	0	0	0	0	1	1	0	0	0	0	0	0	

		P1					P2				P3			
		Tom__2	Jeff_7	Jay__8	Sandy_9	Darrin_11	Ron__1	Frank_3	Boyd_4	Tim__5	John_6	Jerry_10	Ben_12	Arnie_13
P1	Tom__2		0	0	0	1	1	1	0	0	0	0	0	0
	Jeff_7	1		1	1	0	0	0	0	0	0	0	0	0
	Jay__8	1	1		1	0	0	0	0	0	0	0	0	0
	Sandy_9	1	1	1		0	0	0	0	0	0	0	0	0
	Darrin_11	1	0	0	0		1	0	0	0	0	1	0	0
P2	Ron__1	0	0	0	0	0		1	1	1	0	0	0	0
	Frank_3	0	0	0	0	1	1		1	0	0	0	0	0
	Boyd_4	1	0	0	0	0	1	1		0	0	0	0	0
	Tim__5	0	0	0	0	0	1	1	1		0	0	0	0
P3	John_6	0	0	0	0	0	0	0	0	1		0	1	1
	Jerry_10	0	0	0	0	0	1	0	0	0	1		0	0
	Ben_12	0	0	0	0	0	1	0	0	0	1	1		0
	Arnie_13	0	0	0	0	0	0	0	0	1	1	0	0	

	P1	P2	P3
P1	0.55	0.15	0.05
P2	0.10	0.83	0.00
P3	0.00	0.25	0.50

	P1	P2	P3
P1	com	nul	nul
P2	nul	com	nul
P3	nul	nul	com



Socnet.se
Client for direct blockmodeling

- A fast (compiled) CLI-based client for direct blockmodeling
- Version 1.4
- Goodness-of-fit functions:
  - Hamming distances
  - Weighted correlations
- Equivalences:
  - Structural
  - Regular
  - Generalized
- Free-search or specified blockimage

Socnet.se
Client for direct blockmodeling

Socnet code to do direct blockmodeling of Little League TI data:

```
# Set working directory to where your network files are
setwd(C:\Users\pekpi\Dropbox\work\Winter CSS school\data)

# Load the matrix file (and rename it to llti)
loadmatrix(file=little_league_ti.txt, name=llti)

# Specify a non-specified 3-positional blockimage for structural equivalence
bi = blockimage(size=3, pattern=com;nul)

# Initialize the search engine
bminit(network=llti, blockimage=bi, searchtype=exhaustive, method=hamming)

# Start search (can take a while)
bmstart()

# Explore solutions
structures

# To view a Blockmodel (you should have 12), use:
bmview(bm_llti_bi_0)
```

```
> bmview(bm_llti_bi_0)
Blockmodel:
+---------------+
|\  X  |   |XX  |       0_Tom__2
| \  XX|   |   X|       0_John_6
|  X\  |   |X   |       0_Jerry_10
|X X\  |   |X   |       0_Darrin_11
|  XX \|   |X   |       0_Ben_12
|  X   \|  |   X|       0_Arnie_13
+---------------+
|X     |\XX|    |       1_Jeff_7
|X     |X\X|    |       1_Jay__8
|X     |XX\|    |       1_Sandy_9
+---------------+
|      |   |\XXX|       2_Ron__1
|   X  |   |X\X |       2_Frank_3
|X     |   |XX\ |       2_Boyd_4
|      |   |XXX\|       2_Tim__5
+---------------+
Blockimage:
        P0      P1      P2
P0      nul     nul     nul
P1      nul     com     nul
P2      nul     nul     com
Goodness-of-fit: 23 (hamming)
```

Notably, direct blockmodeling finds a better partition (lower goodness-of-fit)

Actually finds <u>two</u> equally optimal solutions (and several isomorphic solutions that are excluded)

Socnet.se
Client for direct blockmodeling

Instead of Hamming, we can use (weighted) correlations as GoF

Then we can't use 'exhaustive', but instead the 'ljubljana' local opt algorithm

```
# Set working directory to where your network files are
setwd(C:\Users\pekpi\Dropbox\work\Winter CSS school\data)

# Load the matrix file (and rename it to llti)
loadmatrix(file=little_league_ti.txt, name=llti)

# Specify a non-specified 3-positional blockimage for structural equivalence
bi = blockimage(size=3, pattern=com;nul)

# Initialize the search engine
bminit(network=llti, blockimage=bi, searchtype=ljubljana, method=nordlund)

# Start search (can take a while with the 'Nordlund' option)
bmstart()

# Explore solutions
structures

# To view the Blockmodel (you should only have one now), use:
bmview(bm_llti_bi_94_0)
```

```
Blockmodel:
+---------------+
|\XX|    | X    |      0_Jeff_7
|X\X|    | X    |      0_Jay__8
|XX\|    | X    |      0_Sandy_9
+---------------+
|   |\ XX|    X |      1_John_6
|   |X\  |X     |      1_Jerry_10
|   |XX\ |X     |      1_Ben_12
|   |X  \|    X |      1_Arnie_13
+---------------+
|   |    |\ XXX |      2_Ron__1
|   |    |X\X  X|      2_Tom__2
|   |    |X \X X|      2_Frank_3
|   |    |XXX\  |      2_Boyd_4
|   |    |X XX\ |      2_Tim__5
|   | X  |XX   \|      2_Darrin_11
+---------------+
Blockimage:
        P0      P1      P2
P0      com     nul     nul
P1      nul     com     nul
P2      nul     nul     com
Goodness-of-fit: 0.5752 (nordlund)
```

When using correlation coefficient, more likely to find singular optimal solution

Arguably easier to compare and interpret the fit than an absolute number of deviations

Socnet.se

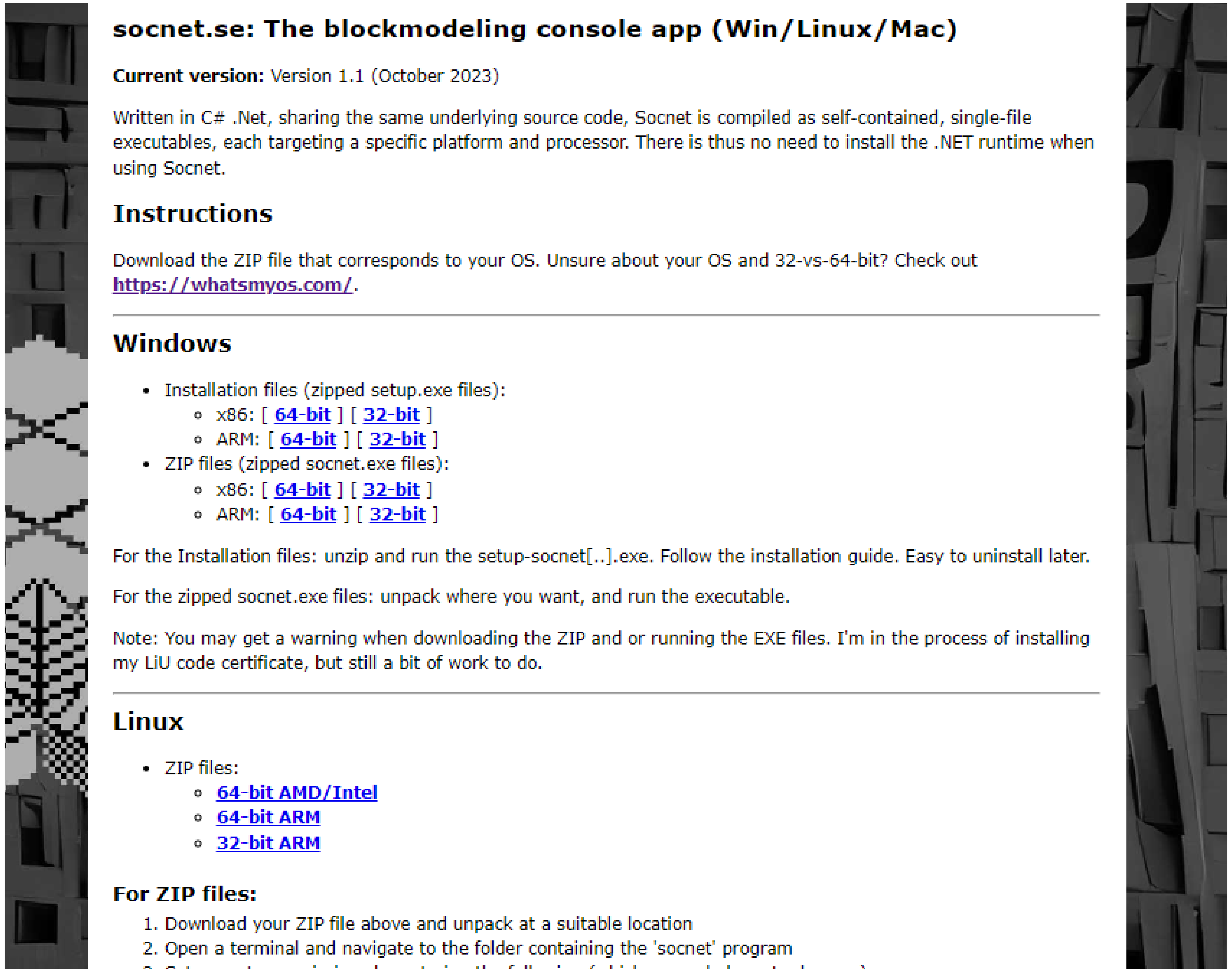
Console/terminal application written in C#/.Net

Download at https://socnet.se/?page=download

Code-signed executable/binaries for Win, Linux, MacOS (setup.exe for Windows)

Although codesigned, could give warning when DL:ing on Chrome (though not Edge)

Full documentation at https://socnet.se/?page=docs



socnet.se: The blockmodeling console app (Win/Linux/Mac)

Current version: Version 1.1 (October 2023)

Written in C# .Net, sharing the same underlying source code, Socnet is compiled as self-contained, single-file executables, each targeting a specific platform and processor. There is thus no need to install the .NET runtime when using Socnet.

Instructions

Download the ZIP file that corresponds to your OS. Unsure about your OS and 32-vs-64-bit? Check out https://whatsmyos.com/.

Windows

- Installation files (zipped setup.exe files):
  - x86: [64-bit] [32-bit]
  - ARM: [64-bit] [32-bit]
- ZIP files (zipped socnet.exe files):
  - x86: [64-bit] [32-bit]
  - ARM: [64-bit] [32-bit]

For the Installation files: unzip and run the setup-socnet[..].exe. Follow the installation guide. Easy to uninstall later.

For the zipped socnet.exe files: unpack where you want, and run the executable.

Note: You may get a warning when downloading the ZIP and or running the EXE files. I'm in the process of installing my LiU code certificate, but still a bit of work to do.

Linux

- ZIP files:
  - 64-bit AMD/Intel
  - 64-bit ARM
  - 32-bit ARM

For ZIP files:

1. Download your ZIP file above and unpack at a suitable location
2. Open a terminal and navigate to the folder containing the 'socnet' program

Afternoon lab

Lab (13-16ish)

- Working with a bunch of classical datasets
  - Baker citation, EIES (t1, t2), Galtung, Hlebec notesharing, Knoke, Little league, Zachary karate club
  - Binary and valued, directional and symmetric
  - Single-layer
  - S50 data (Snijders' Teenage Friends and Lifestyle Study data) https://www.stats.ox.ac.uk/~snijders/siena/s50_data.htm
- R scripts
  - Lab_day_1.Rmd – exploring (also random graphs)
  - Cohesive_subgroups, assortativity (EIES), rewiring

Lab (13-16ish)

- Socnet.se
  - Command-line interface (CLI) console application
  - Download and install from https://socnet.se/
  - Script file: socnet_script.txt
  - Copy and paste (non-commented) commands into Socnet.se client