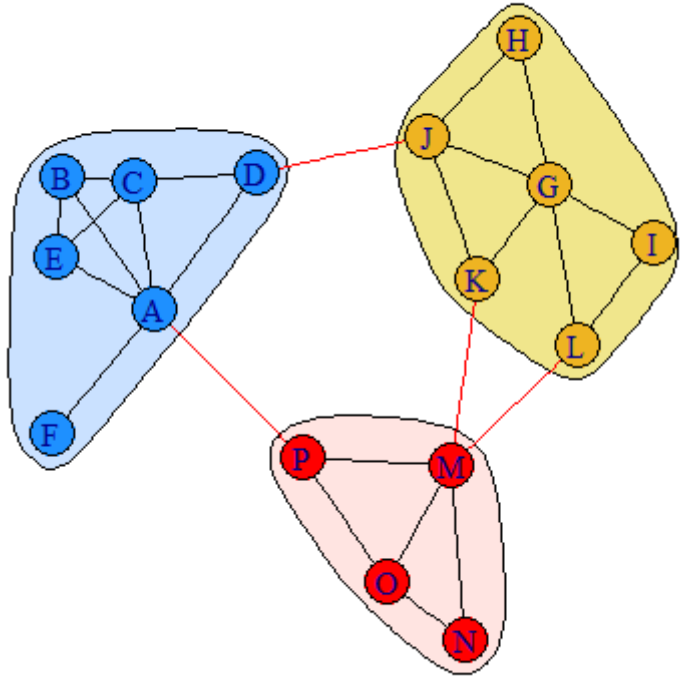
Community Detection



Community Detection Algorithms

- Algorithms aim to identify groups consisting of densely connected nodes

- These groups have high density of connections within groups and fewer connections between groups.

igraph has several built-in community detection algorithms including:

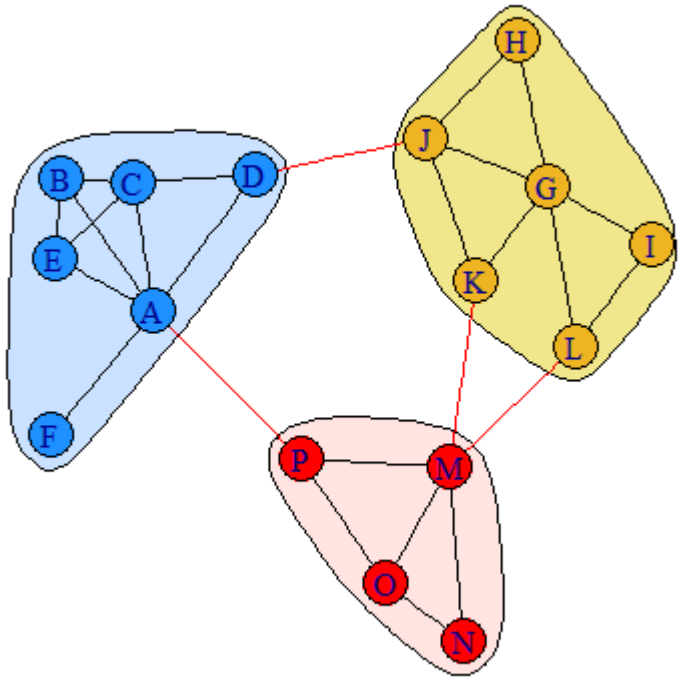
cluster_edge_betweenness(g)
cluster_fast_greedy(g)
cluster_louvain()

Most commonly used in animal behavior

cluster_spinglass(g)
cluster_label_prop(g)
cluster_optimal(g)
cluster_walktrap(g)
cluster_infomap(g) (maybe best for directed graphs)

The 'modMax' library has 38 different algorithms !

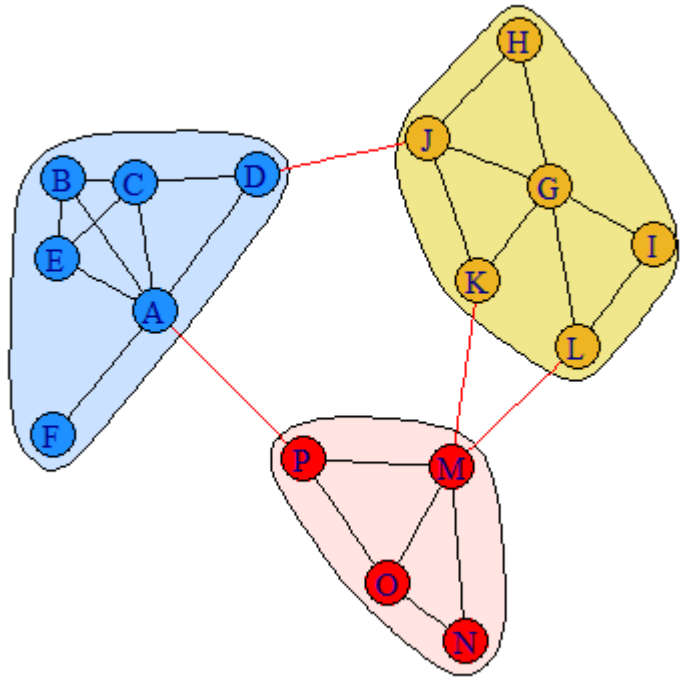
Modularity



- Modularity is used as a measure of goodness-of-fit for community detection
- Modularity = the proportion of edges that occur within communities minus the expected proportion of edges if they were randomly distributed.
- Modularity ranges between -1 & +1
- Networks with high modularity have dense connections within and sparse connections between communities.

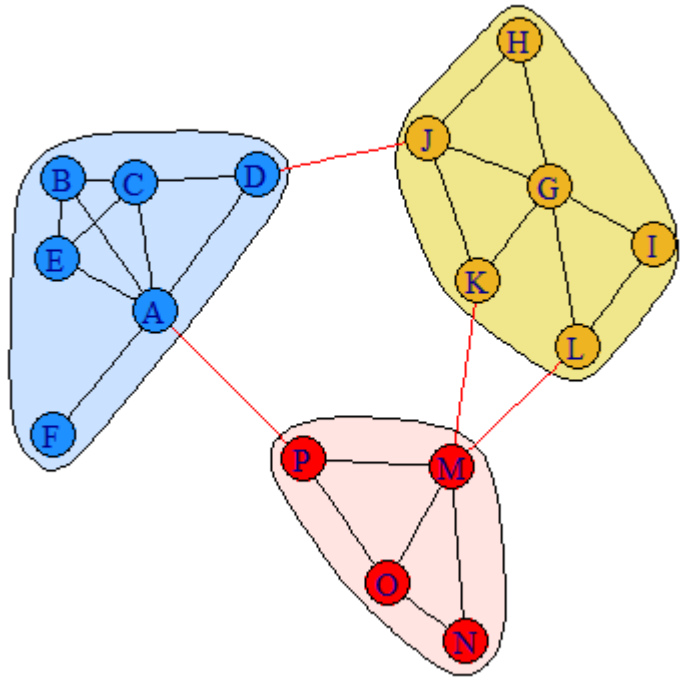
Fast Greedy Algorithm

- Bottom-up approach
- Attempts to optimize modularity “greedily”
- At step 1, every vertex belongs to a separate community
- Communities (of any size) are merged iteratively with each merge leading to the largest increase in modularity
- Algorithm ends when no further increase in modularity is possible
- A fast method
- May have limitations identifying very small communities



Girvan-Newman Edge-Betweenness Algorithm

- Top-down approach, i.e. a divisive algorithm
- At each step the edge with the highest betweenness is removed from the graph.
- Idea is that high edge betweenness edges are those that connect different communities
- Compute the modularity of the graph at each step.
- Use graph with highest value of modularity.
- OK for Weighted and directed networks



Determining Robustness in Community Membership

A critical issue is that these algorithms will always produce a result – they will always generate a community membership

How to generate confidence that these observed community memberships are real ?

Two methods:

1. Bootstrapping (Lusseau et al., 2008)

2. Robust Community Assignment (using assortativity) (Shizuka & Farine 2016)

Bootstrapping Data for Community Detection

Lusseau, D., Whitehead, H., and Gero, S. (2008).
Incorporating uncertainty into the study of animal social networks.
Anim. Behav. 75, 1809-1815.

1. Bootstrap original data with replacement N times. (N is usually at least 1000)

2. Each replicate had the same total number of observations as the original data.

3. For each bootstrap replicate, reassign community membership via the community detection algorithm.

4. From all randomizations, produce a community comembership matrix. Each value represents how many times those two nodes were observed to be in the same community.

5. Use the comembership matrix to determine groupings via e.g. NMDS, tSNE, community detection etc.

Robust Community Detection

Shizuka & Farine (2016)
Measuring the robustness of network community structure using assortativity
Anim. Behav. 112: 237–246.

- Also bootstrap data to construct an n x n matrix M (a comembership matrix of community assignment)

- Construct a co-presence matrix C .1 if both nodes present in replicate, 0 if both not present

- Construct n x n co-membership matrix P where values are proportion of trials where both i/j present that they were in the same matrix

- Calculate rc: the assortativity of P values with the original assignment of communities.

Robust Community Detection

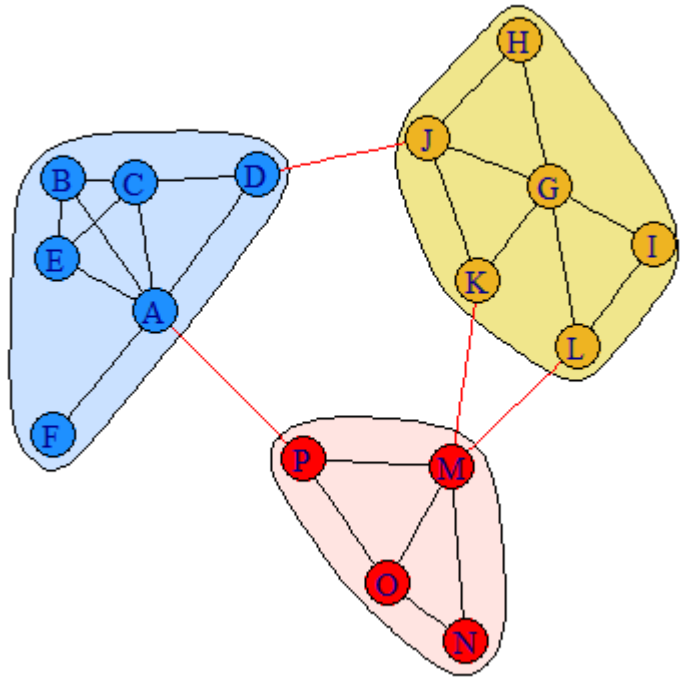
Shizuka & Farine (2016)
Measuring the robustness of network community structure using assortativity
Anim. Behav. 112: 237–246.

rc = 1 if all bootstrap replicates have same community assignments as original

rc = 0 if bootstrap replicate community assignment is same as random networks

rc < 0 if bootstrap replicate community assignment is anti-correlated with original network

Shizuka & Farine recommend values of rc > .5 represent robust community structure



Random Models - Intro

Jackknifing

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

Original

	A	C	D	E	F
A	0	0	0	1	0
C	0	0	1	0	0
D	1	0	0	0	0
E	1	0	1	0	1
F	0	0	0	0	0

Jackknifed

- Jackkifing involves removal of one node at a time, from which the graph is reconstructed
- Generating sampling distributions of our observed network
- Use these to calculate standard errors of observed descriptive statistics
- Can also use these to generate p-values for how ‘unexpected’ our observed descriptive statistics are (but not recommended)

Tom A.B. Snijders & Stephen P. Borgatti, 1999

Node Permutations

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

Original

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

Permuted

- Such permutations would be repeated 1000s of times
- Generating sampling distributions of our observed network
- Use these to calculate standard errors of observed descriptive statistics (can also be used to calculate t-statistics)
- Can also use these to generate p-values for how ‘unexpected’ our observed descriptive statistics are

Tom A.B. Snijders & Stephen P. Borgatti, 1999

One common method is to compare our observed findings to those from standard random graphs:

- Conditional Uniform Graphs (CUGs)
- Classic Random graphs
- ERGMs

Conditional Uniform Graphs (CUGs)

CUGs can be used to test how typical some observed network metric is for a given set of networks. The general routine is as follows:

1. Calculate an observed value of some network metric.

2. Generate many networks that share some characteristic in common with the original network.

3. Compare the observed network metric with the same metric recalculate over all generated networks.

The key question is what characteristics to simulate networks on? There are three default options using the ‘sna’ package:

a) size (number of nodes)
b) number of edges (density)
c) the distribution of dyads (size + density + reciprocity)

As we cannot possibly produce all possible permutations of the graph, we randomly sample from a uniform distribution of graphs that share these properties.

Random Graphs

1. Classic Random Graphs - e.g. Erdos-Renyi (constant probability of an edge between any pair of nodes):

    - large component
    - Poisson degree distribution
    - Low average path length
    - Low clustering

2. Small World Graphs – e.g. Watts & Strogatz (one end of each edge is independently and with probability p rewired to another node)

    - high clustering
    - small distances between nodes

3. Preferential Attachment Models – e.g. Barabasi-Albert model (network grows over time with edges connecting to individuals preferentially, e.g. well connected nodes are preferentially attached to)

    - Discrete time step model
    - Common in many large networks

Randomization Basic Method:

1. Generate the social network from the observed data

2. Calculate and record the test statistic, using conventional statistics such as linear (mixed effect) models on the data from the observed network

3. Randomize the observed data and generate a ‘random’ social network

4. Calculate and record the test statistic, using the exact same model as in 2, but on the random social network

Farine 2014

Why null models ?

- Network data violate independence of parametric statistics by their very nature
- Often what we are studying is the population – not a sample
- Randomization can be used to account for this non-independence
- There are usually other sources of non-independence in data also (e.g. space, time) and these should be accounted for in the randomizations.
- Essentially, the researcher is aiming to produce randomized blocks of pseudo-replicated data. This will enable them to generate a realistic null distribution for use in significance testing.

Null models

- Data sets that are based on observed data in some way
- They may be randomizations of original data (2 main types for static networks: node & data-stream).
- Or they may be based on models based on properties of the original data
- They should strive to maintain constant all other aspects of the data that are not directly relevant to the hypothesis.

Critical issue regarding null models is whether they accurately reflect the structure of the original data – or are they biased in some way?

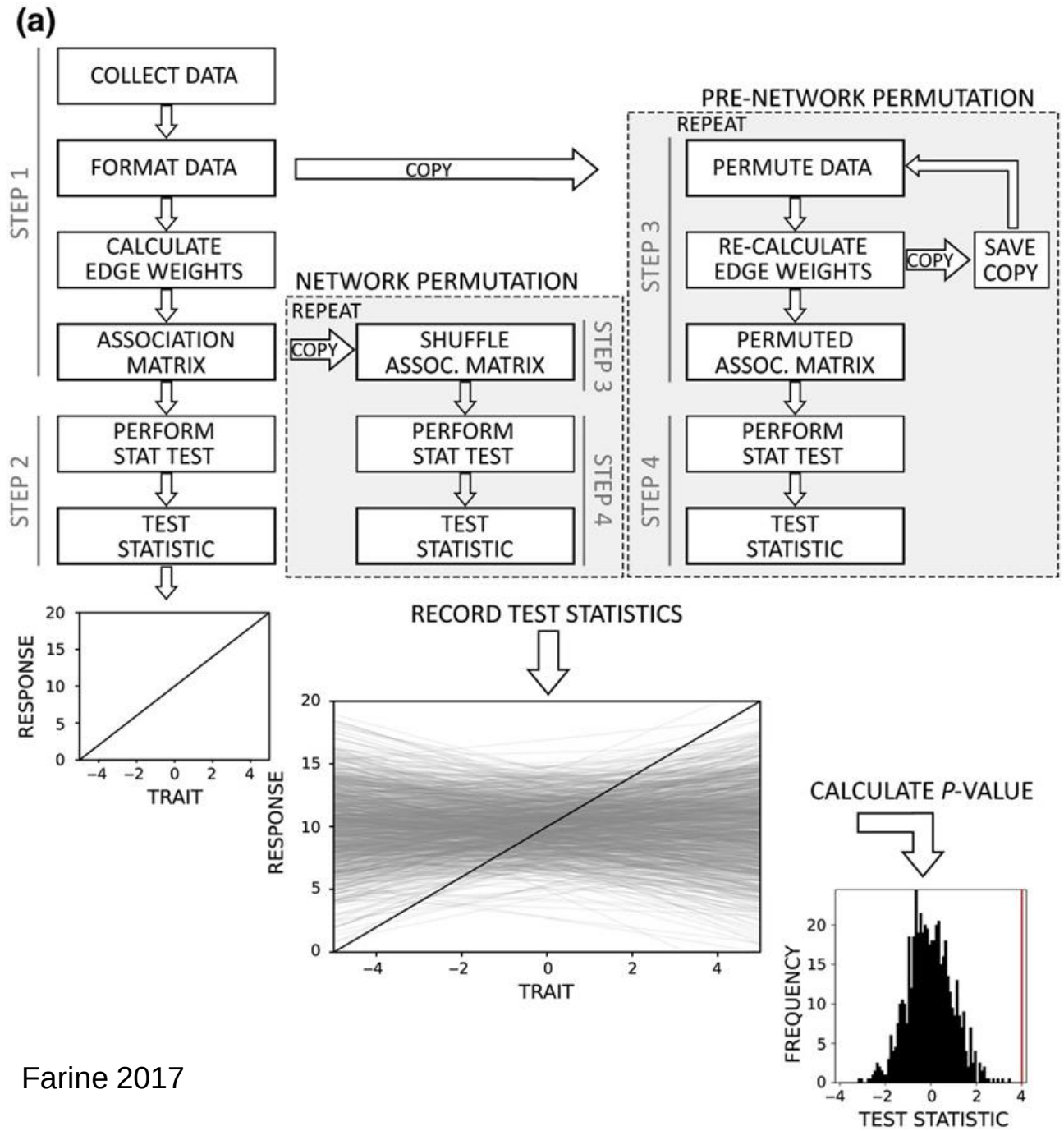
2 types of Data Randomizations in Static Networks based on original data:

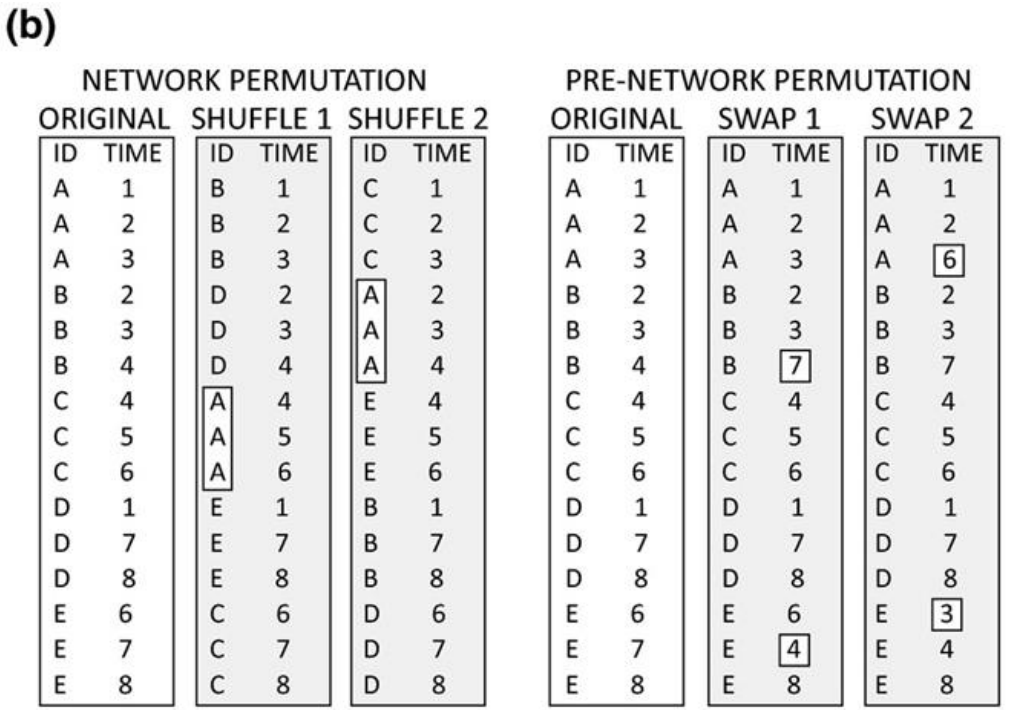
1. Node-based randomizations (Network randomization)

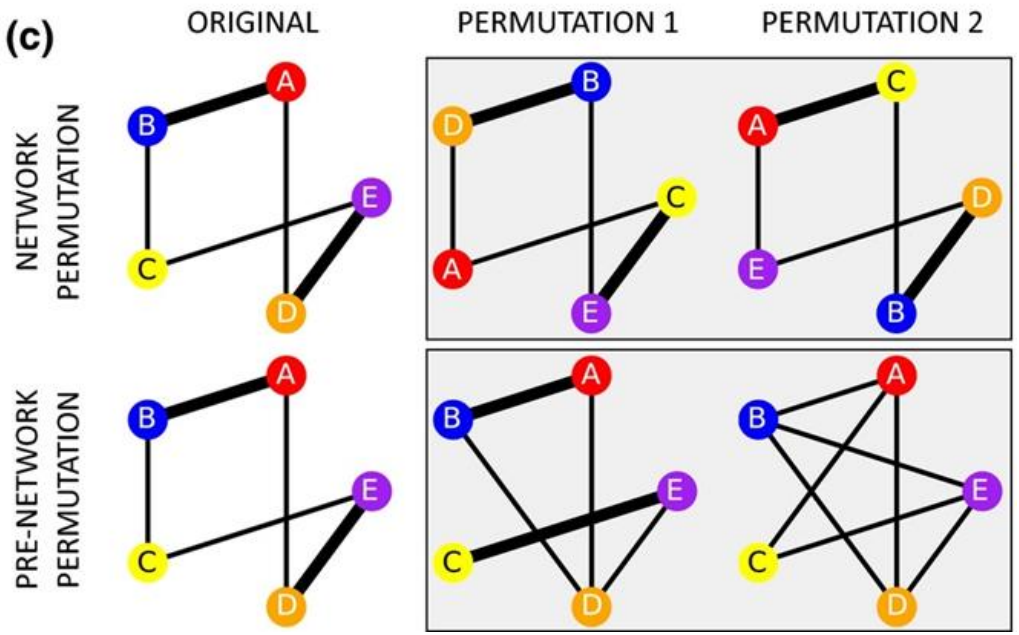
e.g. randomizing attributes of nodes, but maintain the same number of each class. An example would be randomizing gender/sex among nodes. This does assume the observed network is a very good representation of the true network. Farine 2014 has identified that violation of this assumption can lead to higher type I and type II errors.

2. Data Stream-based randomizations (Pre-network randomization)

Sequential swaps between individuals constrained by e.g. time or space. These swaps can occur at the individual level but may also be at the group level.


(a)
STEP 1
COLLECT DATA
FORMAT DATA
CALCULATE EDGE WEIGHTS
ASSOCIATION MATRIX
STEP 2
PERFORM STAT TEST
TEST STATISTIC
COPY
PRE-NETWORK PERMUTATION
REPEAT
STEP 3
PERMUTE DATA
RE-CALCULATE EDGE WEIGHTS
COPY
SAVE COPY
STEP 4
PERMUTED ASSOC. MATRIX
PERFORM STAT TEST
TEST STATISTIC
NETWORK PERMUTATION
REPEAT
COPY
STEP 3
SHUFFLE ASSOC. MATRIX
STEP 4
PERFORM STAT TEST
TEST STATISTIC
RESPONSE
TRAIT
RECORD TEST STATISTICS
CALCULATE P-VALUE
FREQUENCY
TEST STATISTIC


(b)
NETWORK PERMUTATION
ORIGINAL
ID TIME
A 1
A 2
A 3
B 2
B 3
B 4
C 4
C 5
C 6
D 1
D 7
D 8
E 6
E 7
E 8
SHUFFLE 1
ID TIME
B 1
B 2
B 3
D 2
D 3
D 4
A 4
A 5
A 6
E 1
E 7
E 8
C 6
C 7
C 8
SHUFFLE 2
ID TIME
C 1
C 2
C 3
A 2
A 3
A 4
E 4
E 5
E 6
B 1
B 7
B 8
D 6
D 7
D 8
PRE-NETWORK PERMUTATION
ORIGINAL
ID TIME
A 1
A 2
A 3
B 2
B 3
B 4
C 4
C 5
C 6
D 1
D 7
D 8
E 6
E 7
E 8
SWAP 1
ID TIME
A 1
A 2
A 3
B 2
B 3
B 7
C 4
C 5
C 6
D 1
D 7
D 8
E 6
E 4
E 8
SWAP 2
ID TIME
A 1
A 2
A 6
B 2
B 3
B 7
C 4
C 5
C 6
D 1
D 7
D 8
E 3
E 4
E 8


(c)
ORIGINAL
PERMUTATION 1
PERMUTATION 2
NETWORK PERMUTATION
PRE-NETWORK PERMUTATION

Farine 2017

Which test-statistic to use?

Farine's recommendation is to use test statistics that describe data rather than those that represent some departure of the data from a parametric null hypothesis.

e.g. if using a linear model or GLMM to test influence of some metric, best to use coefficients of slopes/effects rather than *t* statistics.

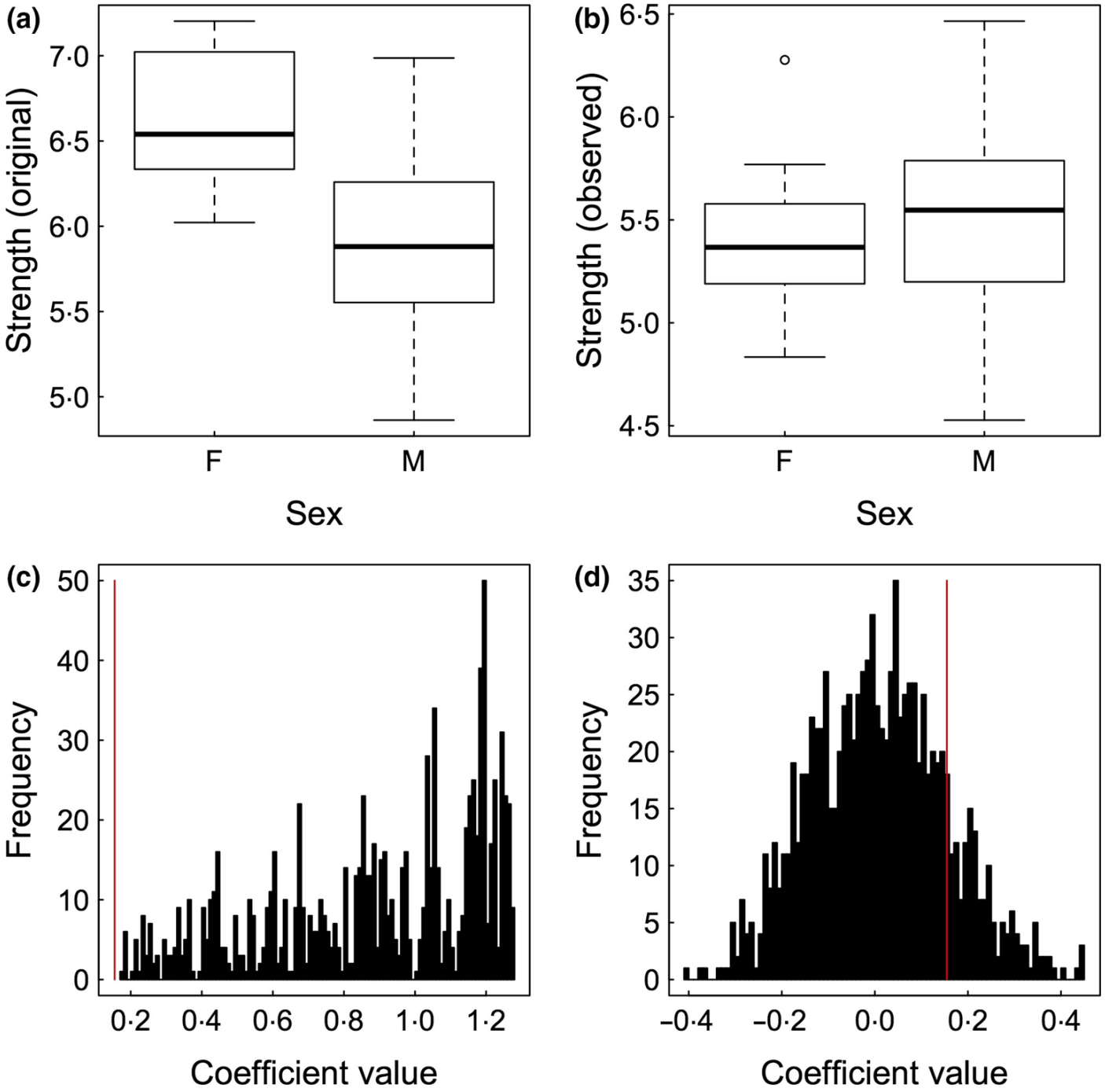
When examining network level metrics – the metrics itself (or e.g. difference between metrics when comparing more than 1 network) can be the test statistic.

How many randomizations are enough ?

Generally, the more the better and a minimum of 1000.

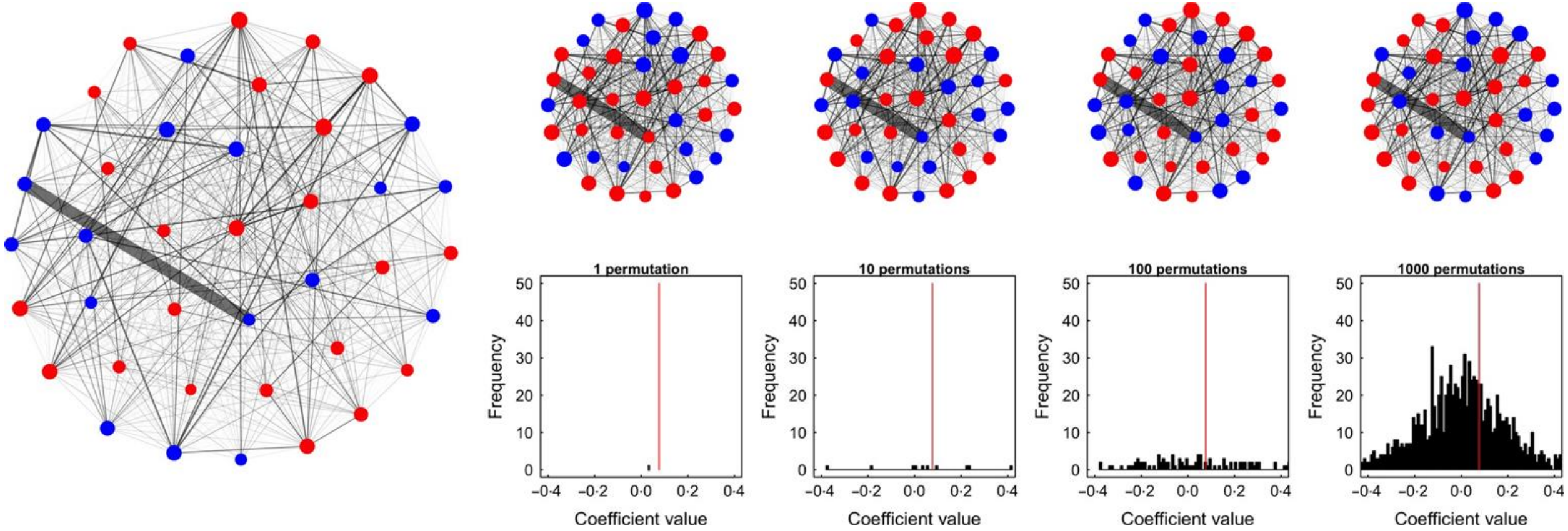
However, best to run with different numbers of randomizations and observe when p-value becomes stabilized (i.e. little difference from run to run of N randomizations).

Data-stream (pre-network) vs Node-based Permutations



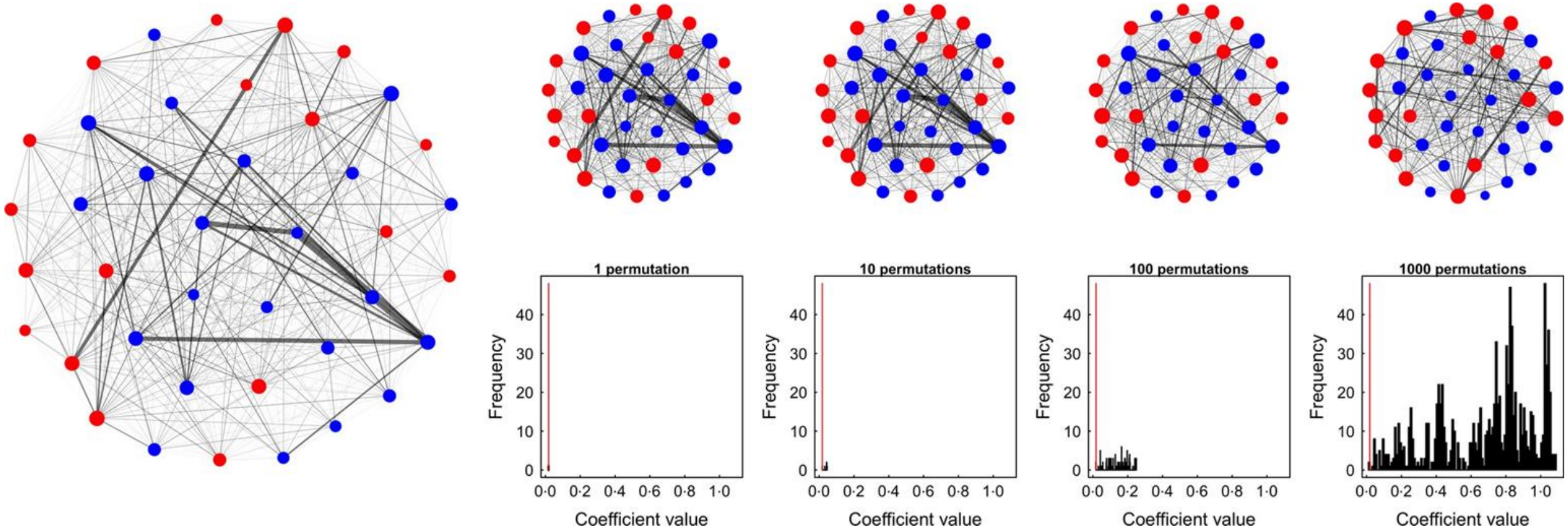
Simulated data demonstrating that pre-network data permutation tests avoid type II errors in hypothesis testing. (a) A dataset is simulated such that females have a higher weighted degree (strength) than males. (b) After removing a random 20% of data from females, the observed data suggest no difference in weighted degree between the sexes. (c) A pre-network data permutation test correctly identifies that the observed coefficient's value (red vertical line) is significantly smaller than that expected by chance (the black histogram). (d) By contrast, a node permutation test does not return the correct result. Both null models used 1000 permutations.

Farine & Whitehead 2015



Example of a node permutation. Data are generated to create a social network (left), but where 20% of female observations are removed. In each permutation (n = 1000), all the node labels in the original network (red = female, blue = male) are randomly re-allocated to new nodes, but the network is kept the same. The same model (Weighted degree ~ Sex) is run for each of the permuted networks, which in this case fails to detect a significant effect (see Fig. 3).

Farine & Whitehead 2015



Example of a pre-network data permutation. Observations of two individuals are swapped between groups, thus in this case only slightly changing the edge structure in the social network with each permutation. Because the swaps are performed incrementally, the network after 1 permutation is very similar to the original network, and thus the coefficient does not change much. However, after many swaps, the coefficient of the model on the permuted networks becomes increasingly different from the coefficient estimated from the observed data, with the final result that females in the observed data have a significantly higher degree relative to males than expected. Note that in this case, the 'random' coefficient values stabilised between values of 0·8 and 1·0, thus providing evidence that a bias is present in the observation data.

Farine & Whitehead 2015

Edge permutations.

It is also possible to generate null models via edge permutations – e.g. randomizing edge attributes between edges in a network.

Although possible, this type of model is not common.

Hypothesis Testing

Test for preferred associations

(Whitehead et al 2005)

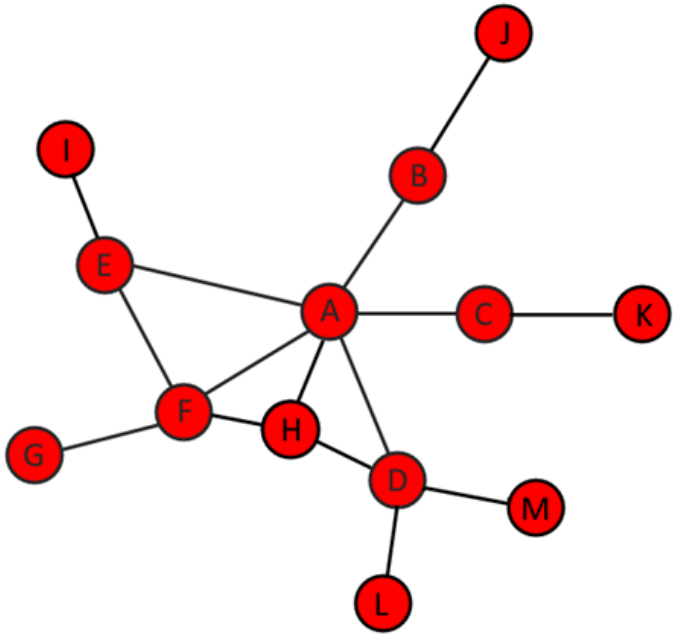
- Take the coefficient of variation (CV) of association indices of a matrix.

- If the network contains more preferred/avoided relationships than expected at random, then the CV of the observed network should be much smaller than the CV of randomized networks.

Using network metrics with lm, glm, glmm etc.

Network data can easily be used as IV or DV in linear, generalized linear and mixed-effect models. In conjunction with randomizations, confidence intervals for observed test-statistics (coefficients) can be generated as can p-values.

What kind of hypotheses might we be studying ?



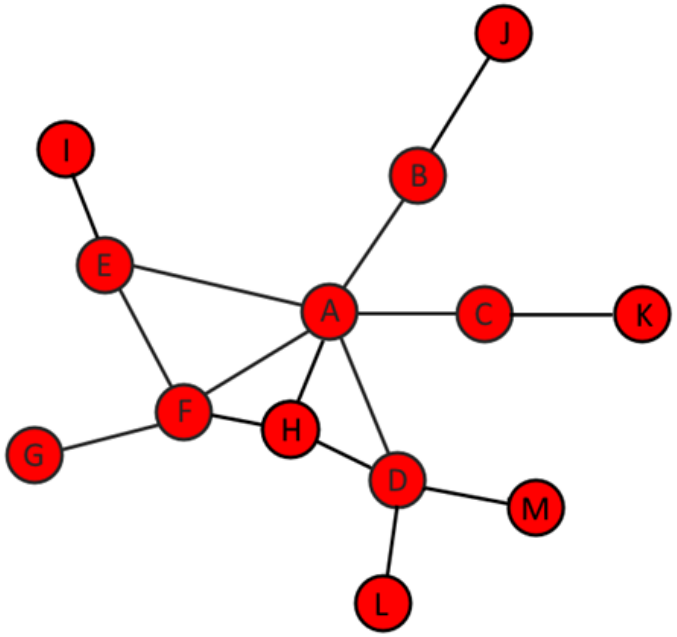
<u>Node (Monadic) x Dyadic Hypotheses</u>

e.g. does gender in a network affect who is friends with whom?

e.g. are individual's attitudes affected by whom they interact with ?

Note: In each of these the IV and DV are different. (gender node-IV, friendships dyad-DV); (interactions dyad-DV, attitudes node-IV)

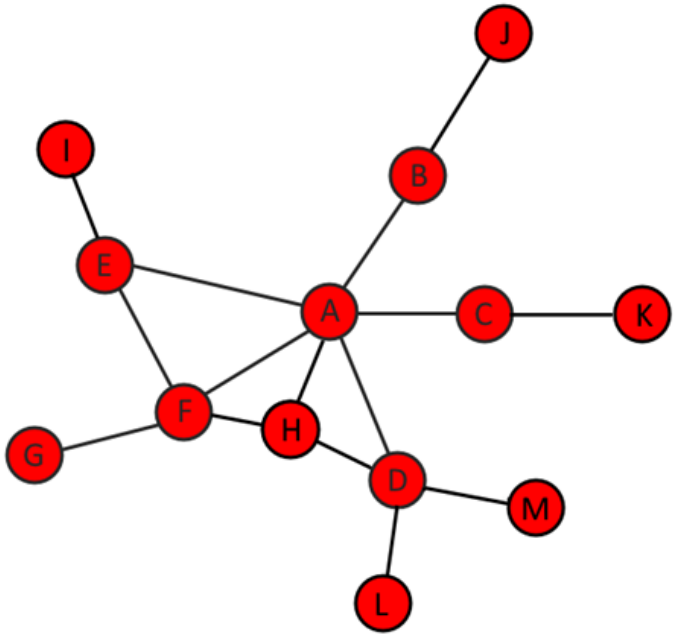
What kind of hypotheses might we be studying ?



Dyadic Hypotheses

e.g. are relationships in one network also observed in another network of the same actors ? - are these relationships similar in strength ?

What kind of hypotheses might we be studying ?



Group / Network Level Hypotheses

e.g. IV may be some measure of social structure of network, DV may be e.g. productivity of group.

e.g. is the observed graph measure particularly unusual for a graph of its size, density etc.?

QAP / Mantel Test

- Comparing the correlation / inter-relationship between two matrices.

- e.g. Does a matrix of association indices correlate with a matrix of genetic relatedness?

- e.g. Does a matrix of grooming interactions correlate with a matrix of aggressive interactions ?

- e.g. Does a matrix of friendship nominations correlate with a matrix of gender co-membership ?

The Mantel (Mantel 1967) and QAP tests calculate the correlation between the off-diagonal elements of each matrix. This value is compared against the distribution of correlations obtained via node-based permutations.

MRQAP

- As QAP is to correlation, MRQAP is to linear regression

- e.g. Can a matrix of association indices be predicted by a matrix of genetic relatedness and a matrix of gender co-membership ?

- e.g. Can a matrix of grooming interactions be predicted by a matrix of aggressive interactions and a matrix of age differences ?

There are different methods for MRQAP which both involve permutation based approaches. The standard MRQAP approach evaluates all fixed effects simultaneously. A more recent method (double-semi-partialling – DSP) tests the effect of each fixed effet whilst covarying for the other fixed effects. Effect sizes can be partial correlation coefficients.

MRQAP – Logistic Regression

If the DV matrix in a MRQAP is binary in nature, then we can perform a logistic MRQAP

How reliable is my null model ?

- QAP & MRQAP both are based on node permutations

- Simulating networks based on properties of original network (e.g. ERGM) are by definition based on the original network

- These may have limitations if the format in which data were collected is not taken into account in the null model

Consider data-stream randomizations

- e.g. to test if two networks are different in some measure e.g. reciprocity
- The observed statistic would be the differene in the measure
- Then both network would be randomized and this differene recalculated for the randomized networks
- P-value is calculated by comparing the observed difference to the distribution of possible differences from randomized data