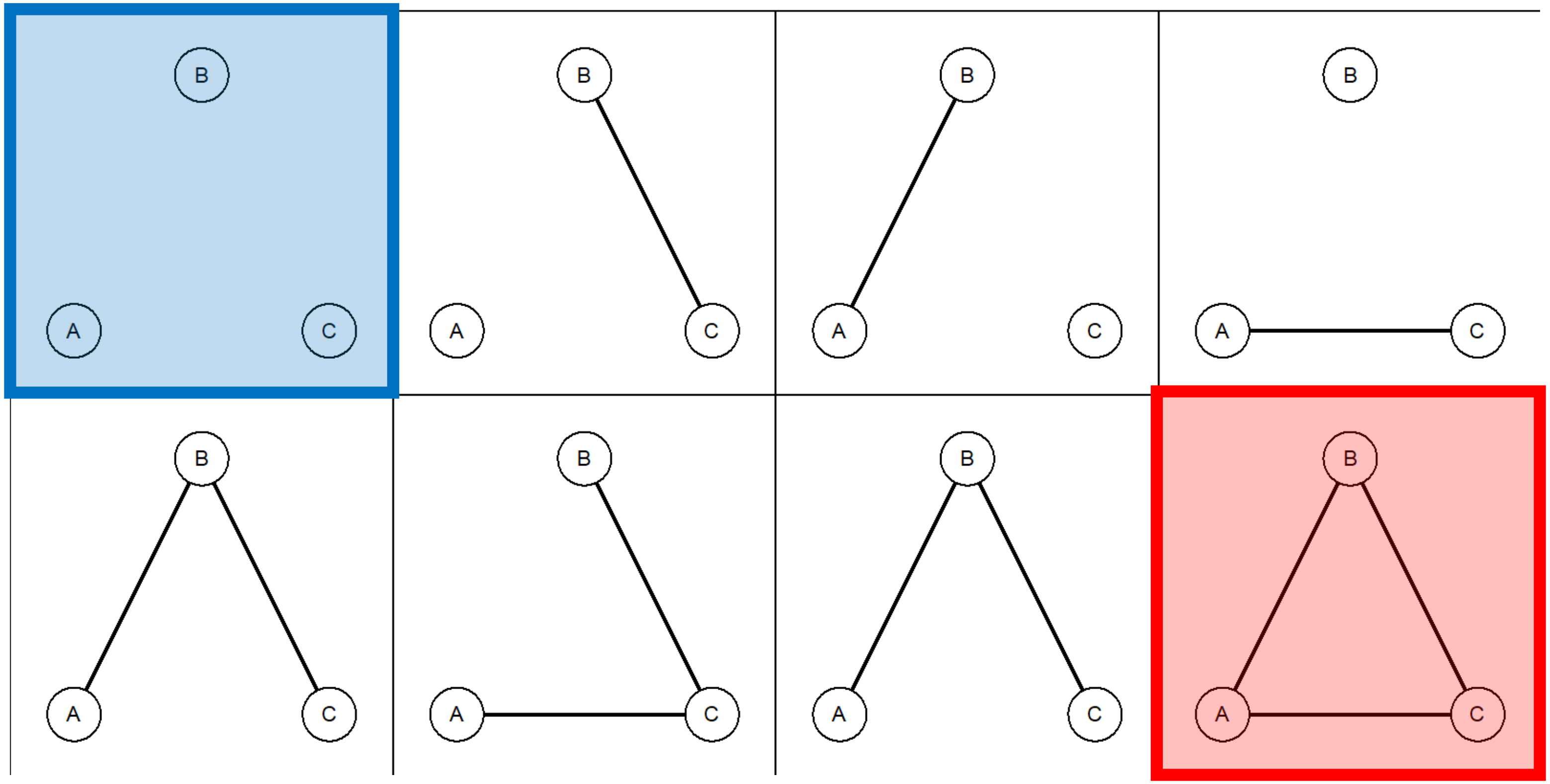
Model Selection

Part 1: Introduction

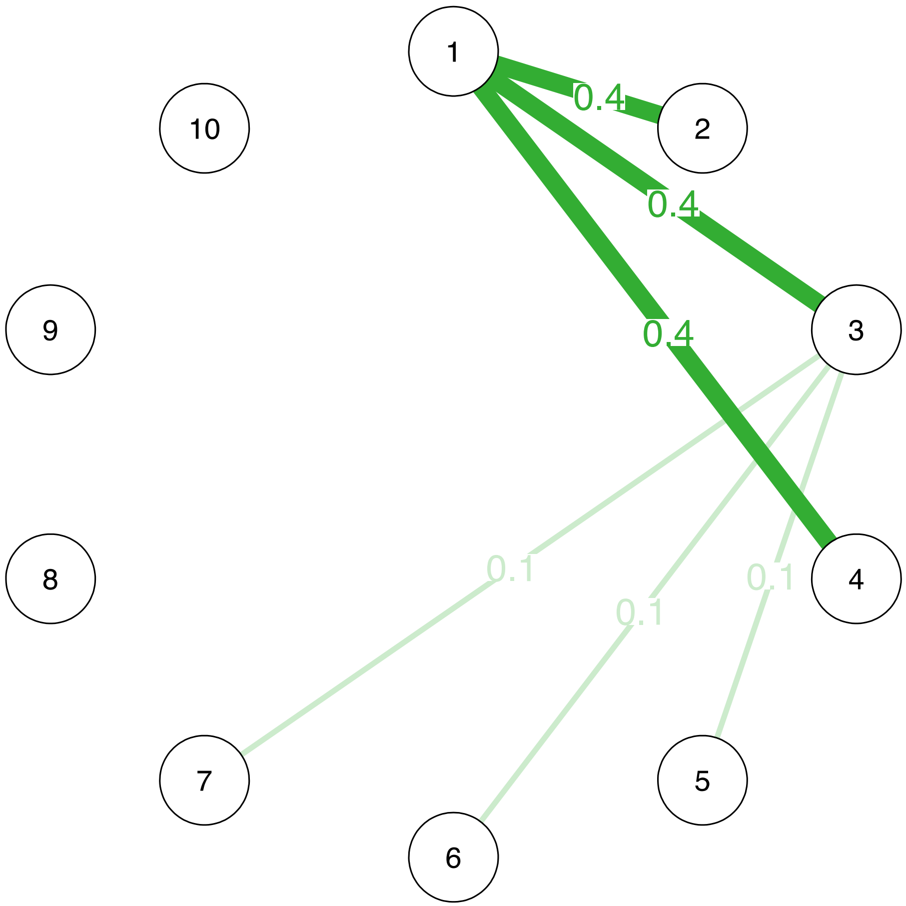
The problem of model selection

A simple (independence) model

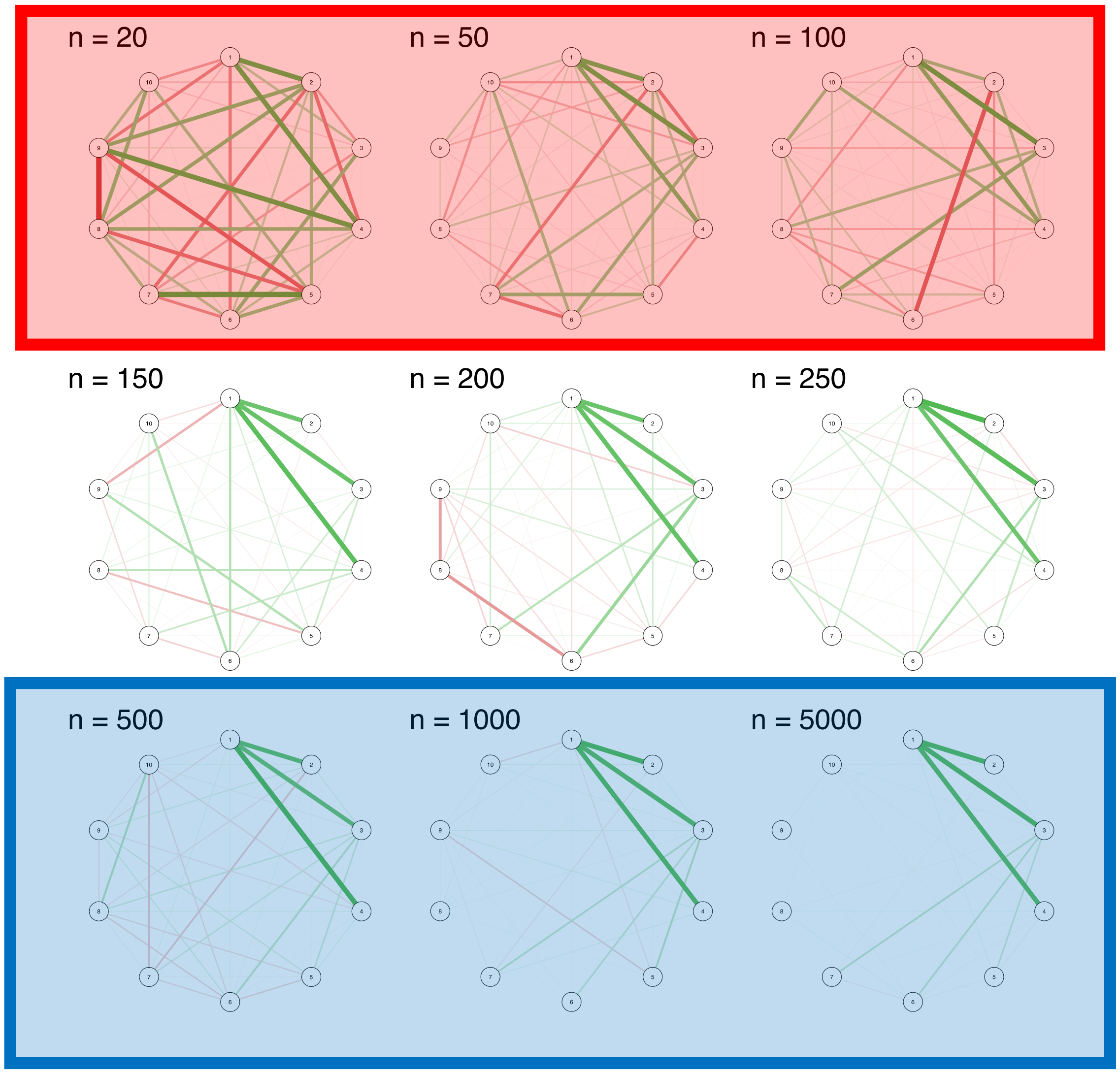


A complicated (saturated) model

Let's sample data from a multivariate Gaussian with the following partial correlations:



And estimate the model with different numbers of observations n:



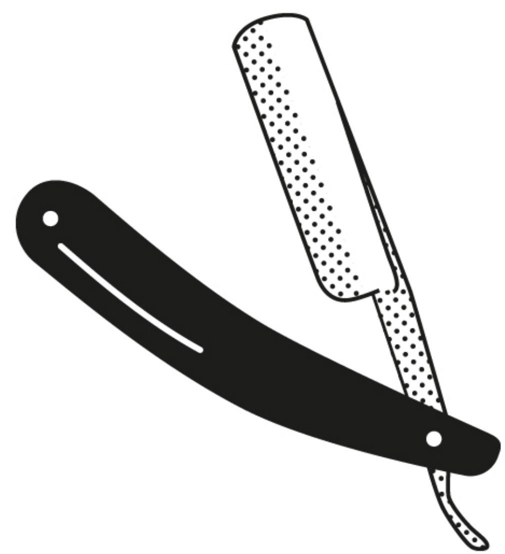
At low n: terrible inference!

At higher n, more accurate pictures, but still many false (vague) edges

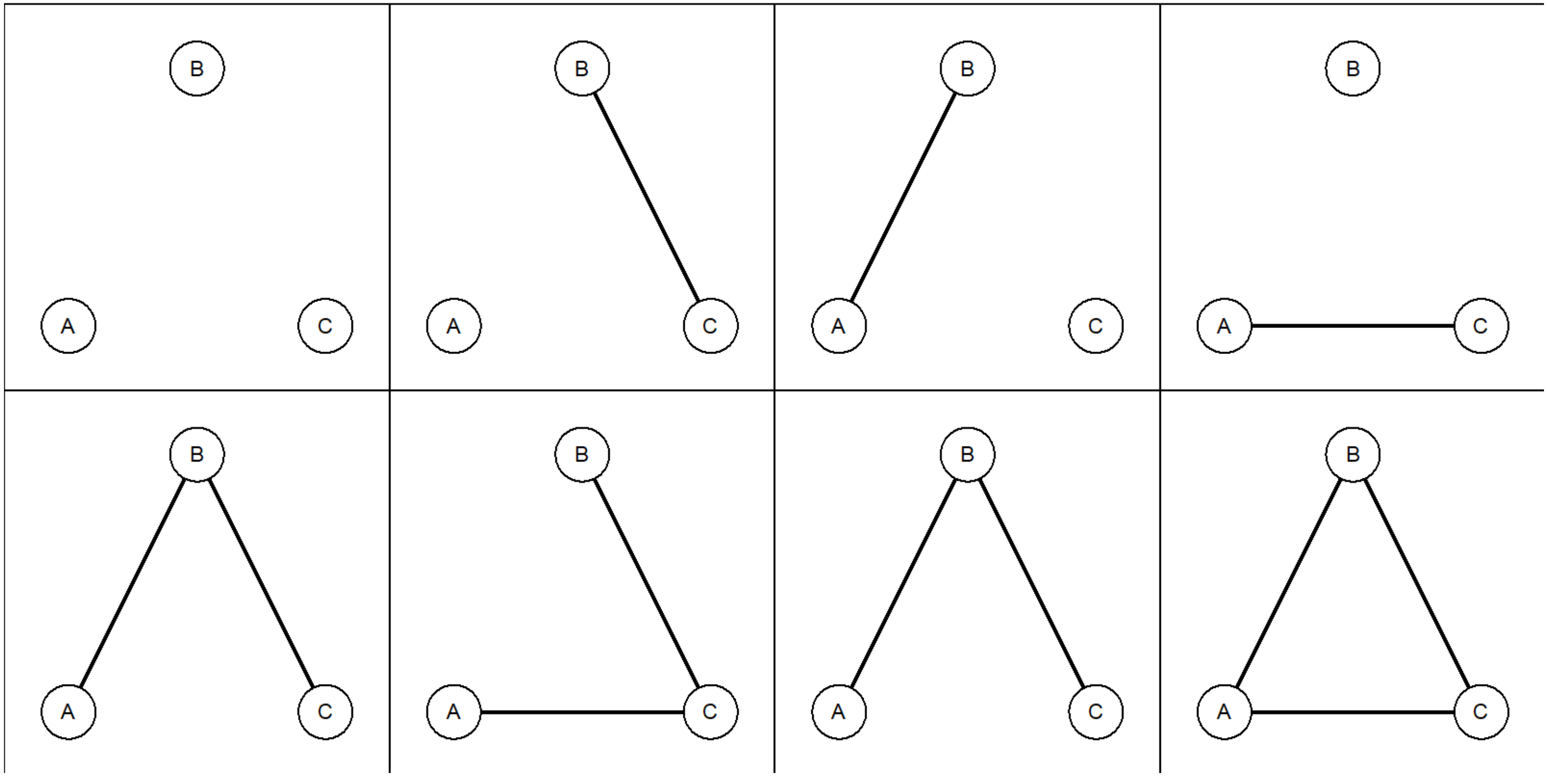
No model selection! All models are equally complex (all edges are estimated)!

Model Selection

- Occam's Razor: when comparing equally performing models, we prefer the simplest one!
- In networks: we want to remove edges (simplify the model) whenever possible
- Because of this: edges that **are** included in the network are **substantial**
    - At least, so we hope
- Many model selection algorithms have been proposed, not clear yet which one is the best!



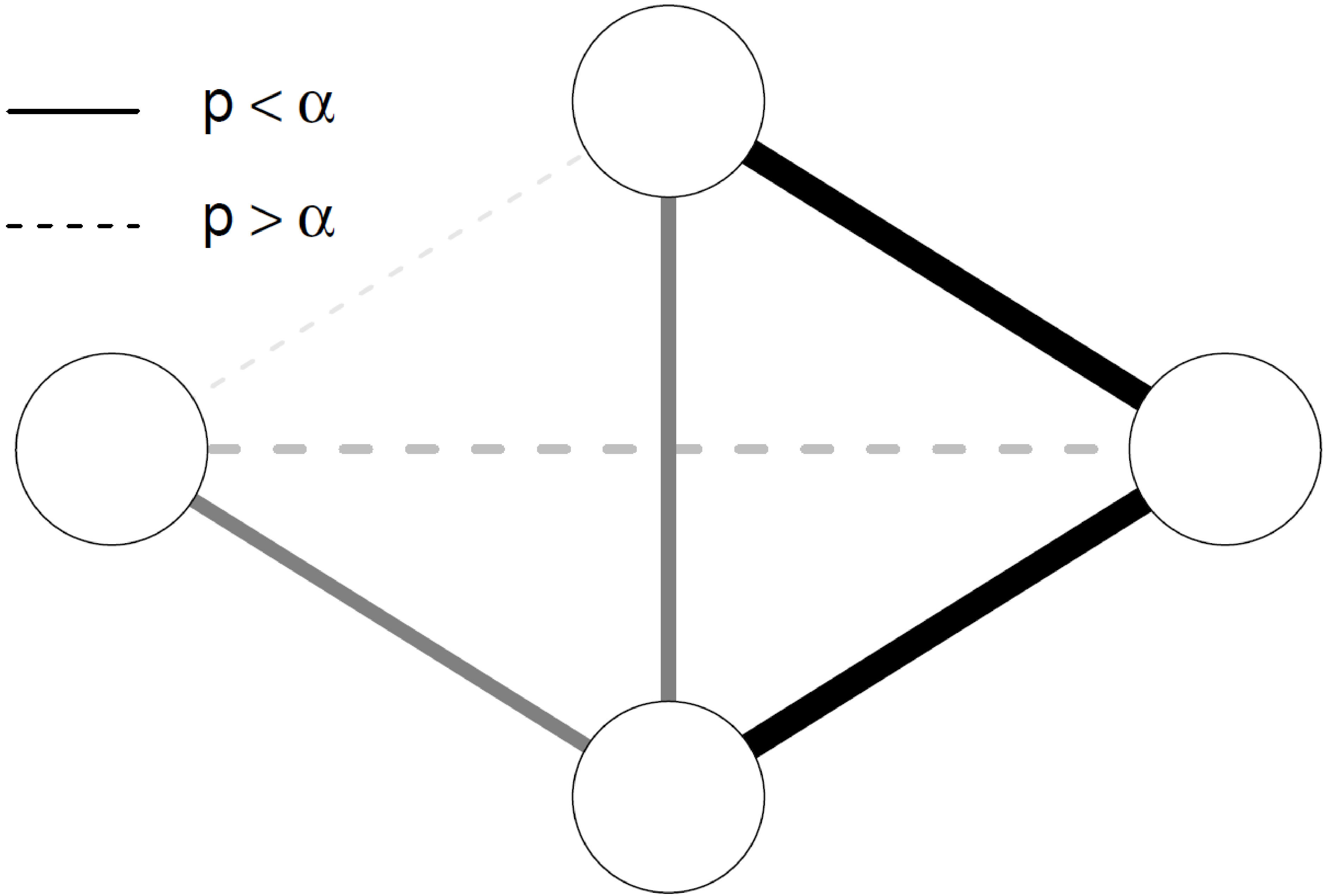
How to select the best model?



1. Thresholding
2. Pruning
3. Regularization
4. Model search

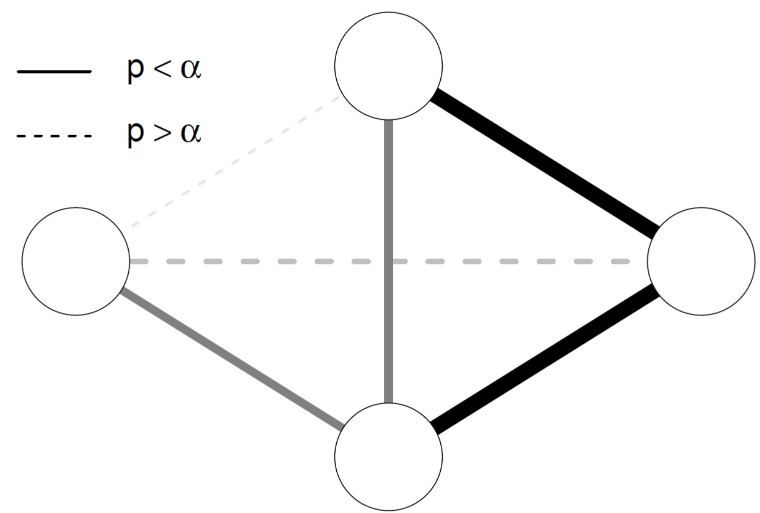
Model Selection

Part 2: Thresholding & Pruning

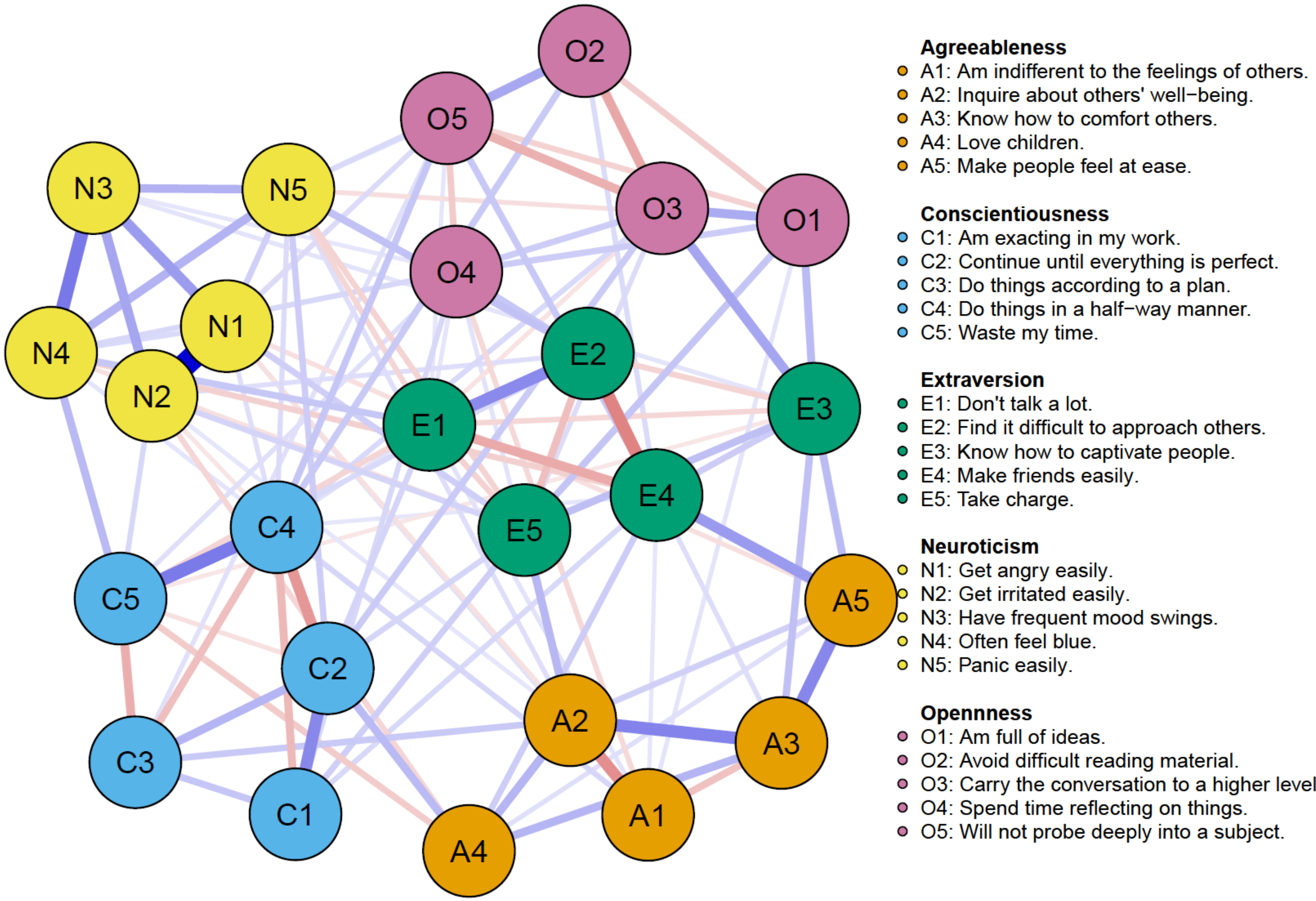

p < α
p > α

Thresholding & Pruning

- Thresholding: hide edges based on some criterion
- Pruning: remove edges based on some criterion and **re-estimate** a model with those edges fixed to zero
- Commonly based on (bootstrapped) significance values, false discovery rates, or Bayes factor
- In node-wise estimation, two values per edge
  - AND-rule / OR-rule

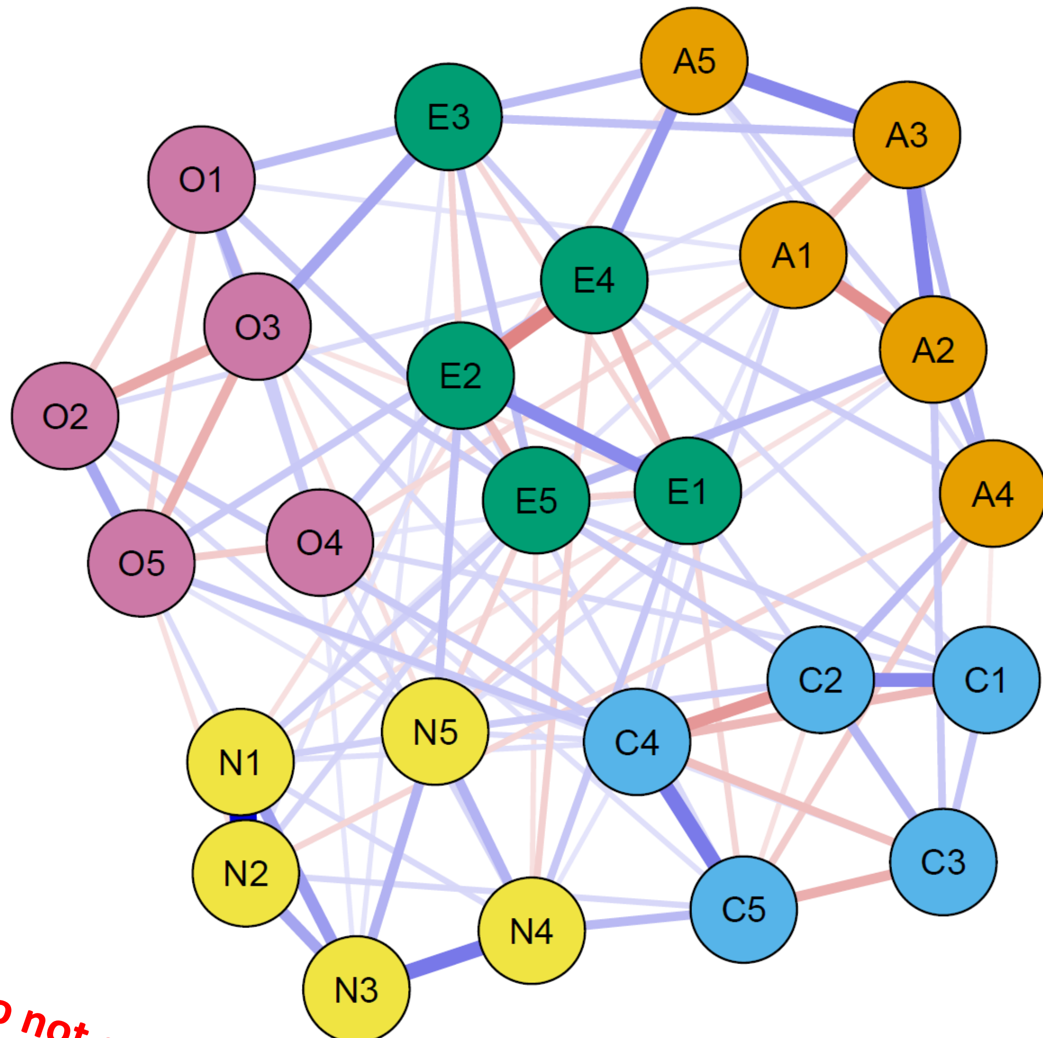


Thresholding: significance values (bootnet)




```
estimateNetwork(bfiSub, default = "pcor",
          threshold = "sig", alpha = 0.01)
```

Thresholding: Bootstrapped significance values (bootnet)



Agreeableness
- A1: Am indifferent to the feelings of others.
- A2: Inquire about others' well−being.
- A3: Know how to comfort others.
- A4: Love children.
- A5: Make people feel at ease.

Conscientiousness
- C1: Am exacting in my work.
- C2: Continue until everything is perfect.
- C3: Do things according to a plan.
- C4: Do things in a half−way manner.
- C5: Waste my time.

Extraversion
- E1: Don't talk a lot.
- E2: Find it difficult to approach others.
- E3: Know how to captivate people.
- E4: Make friends easily.
- E5: Take charge.

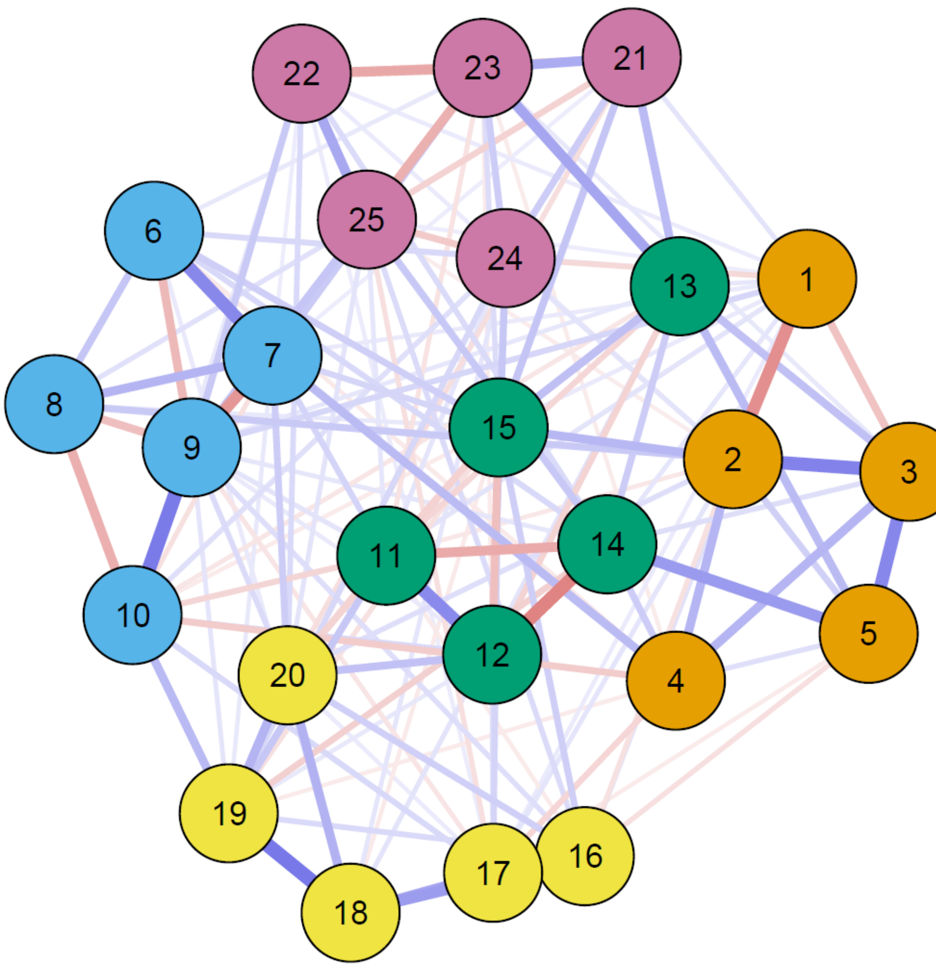
Neuroticism
- N1: Get angry easily.
- N2: Get irritated easily.
- N3: Have frequent mood swings.
- N4: Often feel blue.
- N5: Panic easily.

Opennness
- O1: Am full of ideas.
- O2: Avoid difficult reading material.
- O3: Carry the conversation to a higher level.
- O4: Spend time reflecting on things.
- O5: Will not probe deeply into a subject.

Do not combine this with other model selection methods!

```
net <- estimateNetwork(bfiSub, default = "pcor")
boots <- bootnet(net, nCores = 8)
bootThreshold(boots, alpha = 0.01)
```

Thresholding: Credibility interval (BGGM)



Agreeableness
- 1: Am indifferent to the feelings of others.
- 2: Inquire about others' well−being.
- 3: Know how to comfort others.
- 4: Love children.
- 5: Make people feel at ease.

Conscientiousness
- 6: Am exacting in my work.
- 7: Continue until everything is perfect.
- 8: Do things according to a plan.
- 9: Do things in a half−way manner.
- 10: Waste my time.

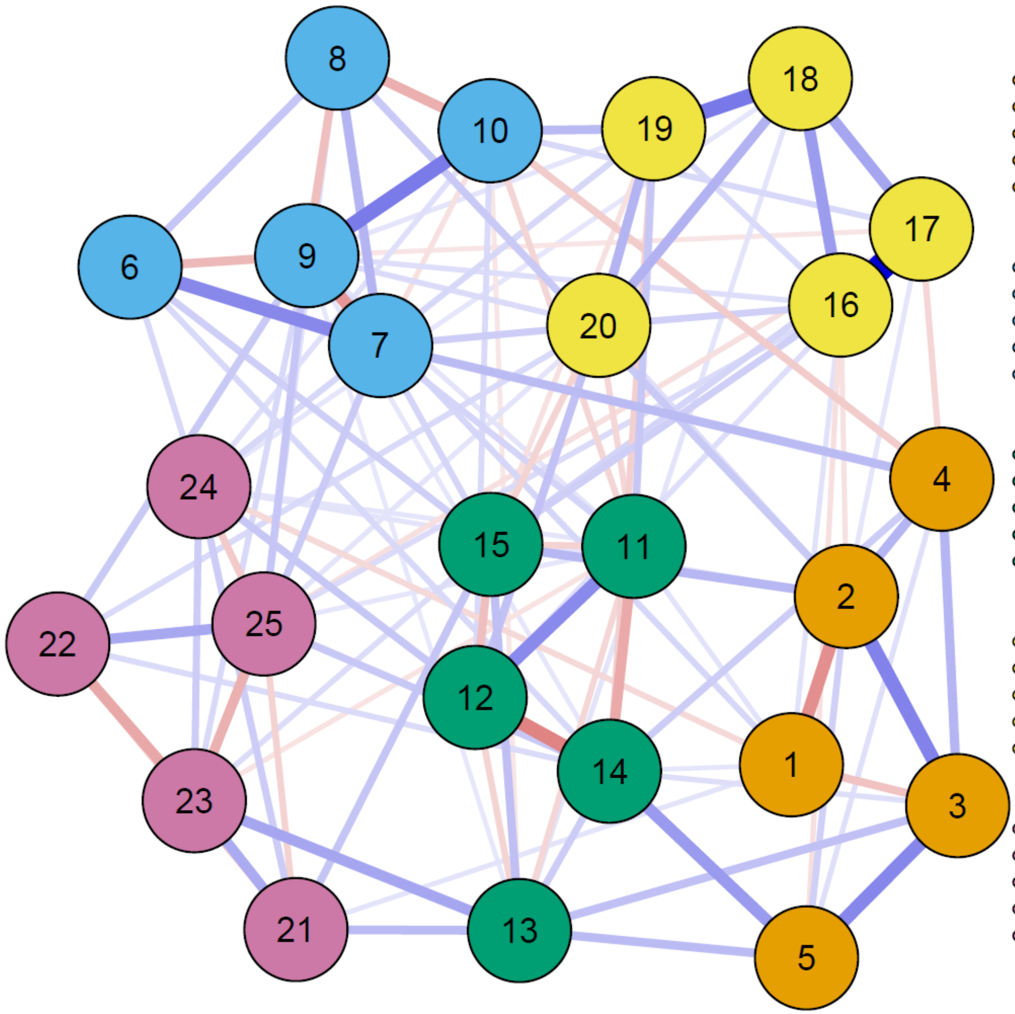
Extraversion
- 11: Don't talk a lot.
- 12: Find it difficult to approach others.
- 13: Know how to captivate people.
- 14: Make friends easily.
- 15: Take charge.

Neuroticism
- 16: Get angry easily.
- 17: Get irritated easily.
- 18: Have frequent mood swings.
- 19: Often feel blue.
- 20: Panic easily.

Opennness
- 21: Am full of ideas.
- 22: Avoid difficult reading material.
- 23: Carry the conversation to a higher level.
- 24: Spend time reflecting on things.
- 25: Will not probe deeply into a subject.

```
fit <- estimate(bfiSub)
net <- select(fit)
net$pcor_adj
```

Thresholding: Bayes factor (BGGM)



Agreeableness
- 1: Am indifferent to the feelings of others.
- 2: Inquire about others' well−being.
- 3: Know how to comfort others.
- 4: Love children.
- 5: Make people feel at ease.

Conscientiousness
- 6: Am exacting in my work.
- 7: Continue until everything is perfect.
- 8: Do things according to a plan.
- 9: Do things in a half−way manner.
- 10: Waste my time.

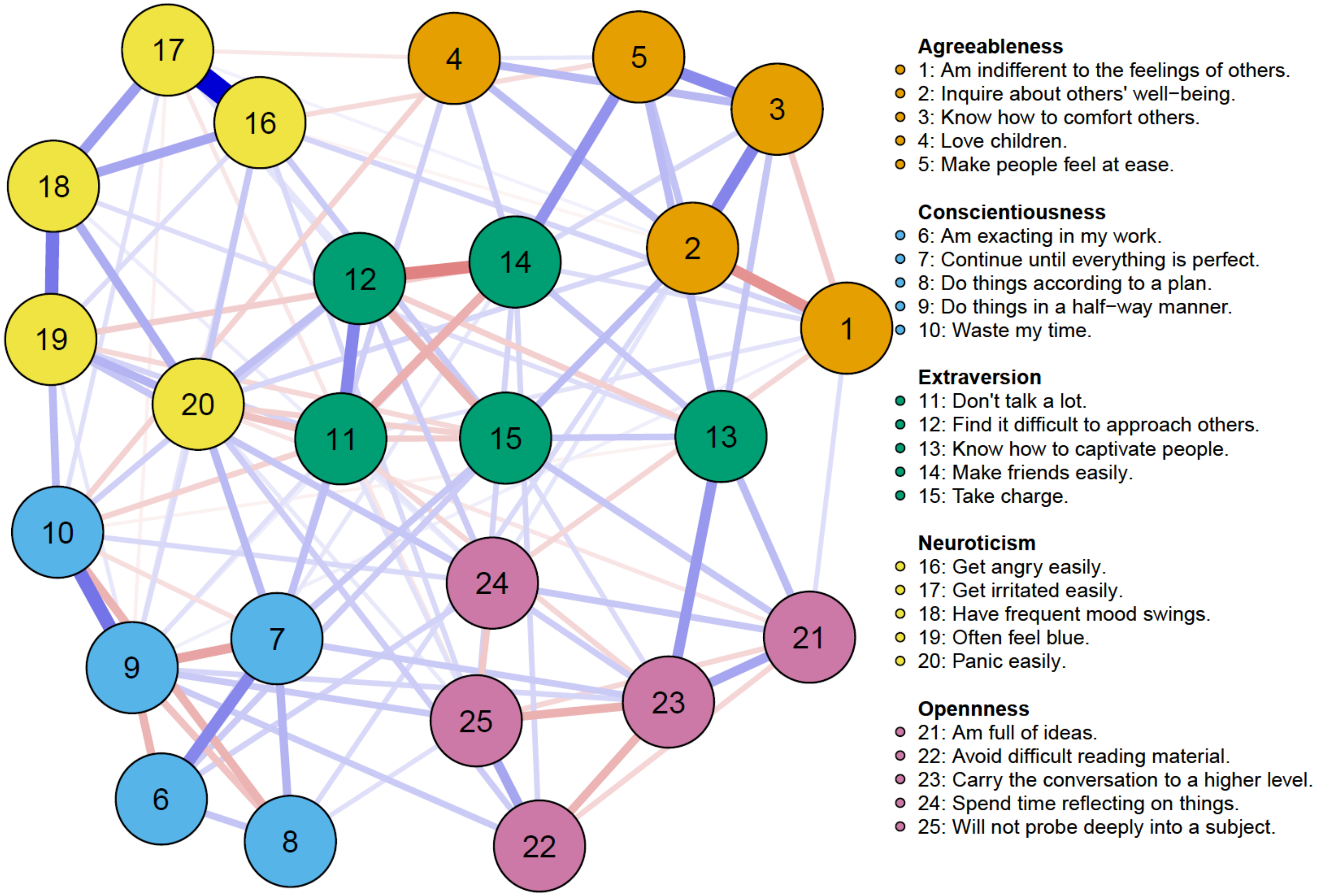
Extraversion
- 11: Don't talk a lot.
- 12: Find it difficult to approach others.
- 13: Know how to captivate people.
- 14: Make friends easily.
- 15: Take charge.

Neuroticism
- 16: Get angry easily.
- 17: Get irritated easily.
- 18: Have frequent mood swings.
- 19: Often feel blue.
- 20: Panic easily.

Openness
- 21: Am full of ideas.
- 22: Avoid difficult reading material.
- 23: Carry the conversation to a higher level.
- 24: Spend time reflecting on things.
- 25: Will not probe deeply into a subject.

```
fit <- explore(bfiSub)
net <- select(fit)
net$pcor_mat_zero
```

Pruning: significance (psychonetrics)




```
mod <- ggm(bfiSub) %>% runmodel %>% prune(alpha = 0.01)

net <- getmatrix(mod, "omega")
```

Thresholding & Pruning

+

- Very fast
- Fixed false positive rate
  - Conservative
- Non-biased estimates

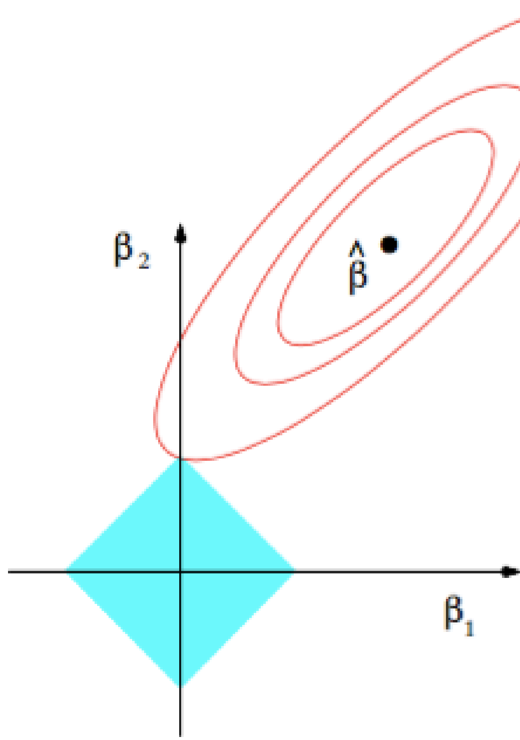
-

- No model selection (thresholding)
- Can lack sensitivity (power)
- Potentially prominent false edges
  - Especially in Ising model
- Not always possible

Model Selection

Part 3: LASSO Regularization

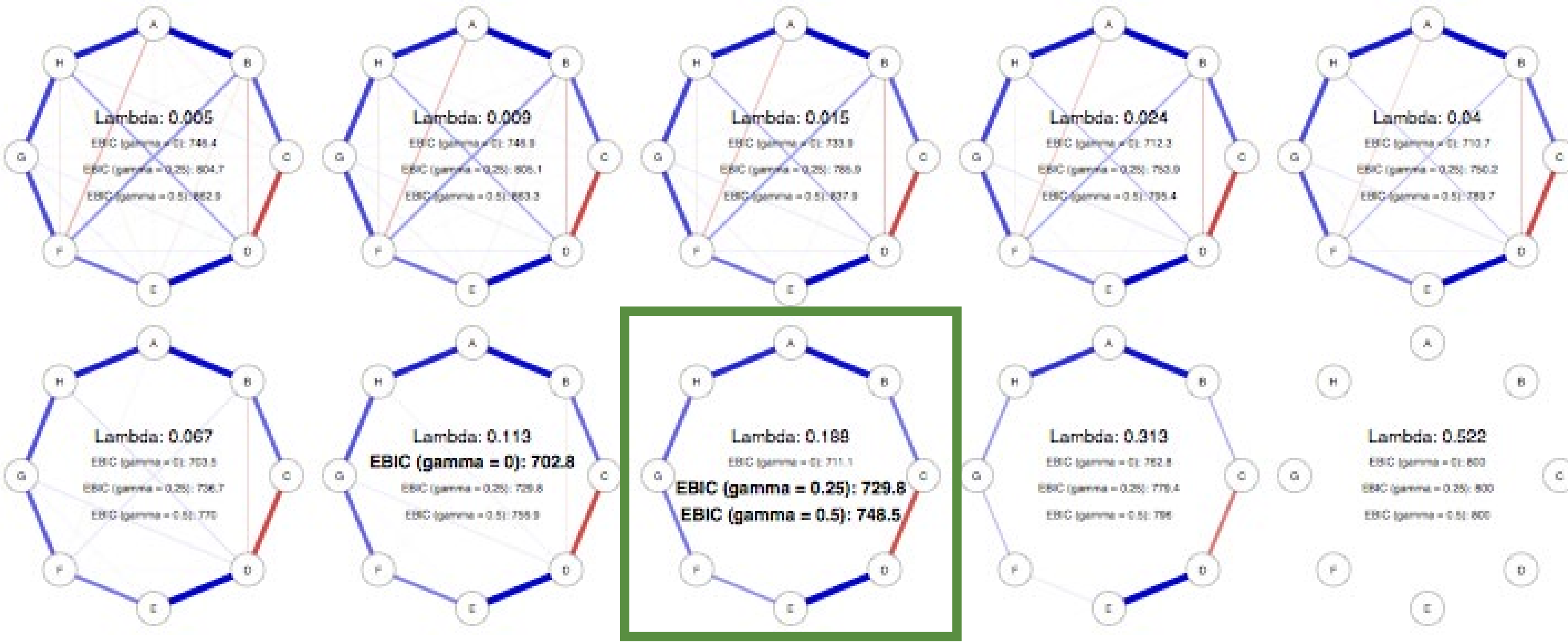
Step 1: Estimate parameters in bounded parameter space...



The size of the box is controlled by tuning parameter λ (higher = smaller box)

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*

Step 2: Vary λ and select the best model

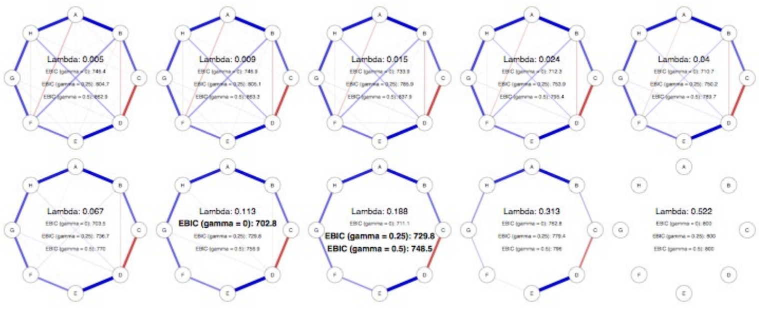


EBIC uses a hypertuning parameter γ (gamma) that needs to be set manually.
Typically between 0 (less conservative) and 1 (more conservative)

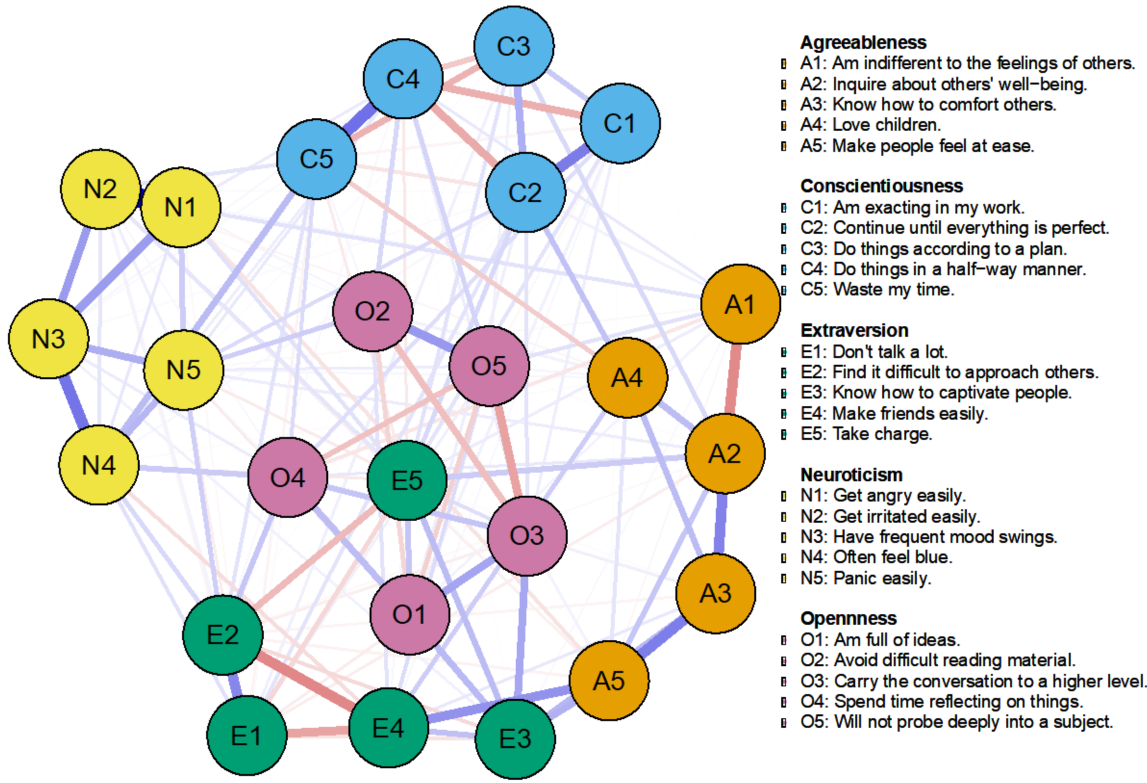
LASSO Regularization

- LASSO regularization with EBIC model selection can be used to estimate network parameters together with network structure
  - For nodewise (logistic) regressions, an AND or OR rule can be used to select edges if one or both regression parameters are non-zero (IsingFit / mgm)
  - For the GGM, the glasso algorithm directly applies LASSO to the precision matrix (EBICglasso)

- Alternative to EBIC, cross-validation can be used



Estimating a partial correlation network (GGM) with model selection – glasso with EBIC model selection




```
estimateNetwork(bfiSub,
      default = "EBICglasso",
      corMethod = "spearman"))
```

For binary data, use default = "IsingFit"

LASSO regularization

+

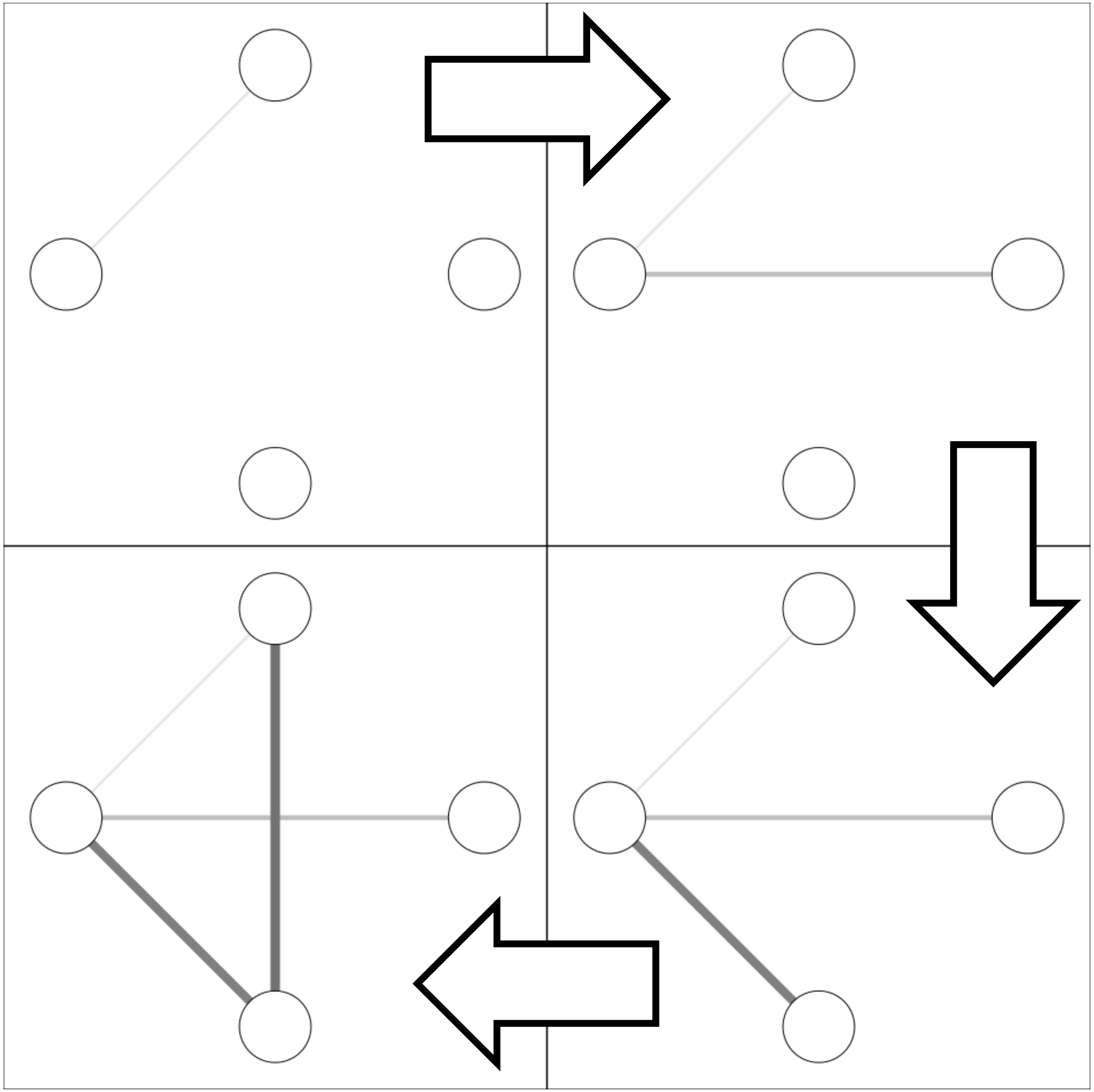
- Very fast and sensitive to detecting edges
- Retrieves a structure on low sample sizes
- Can create a clearer picture by pulling weak edges to zero

-

- Poor performance in high sample sizes
- Strongly reliant on assumption of sparsity
- No fixed false positive rate (less conservative)

Model Selection

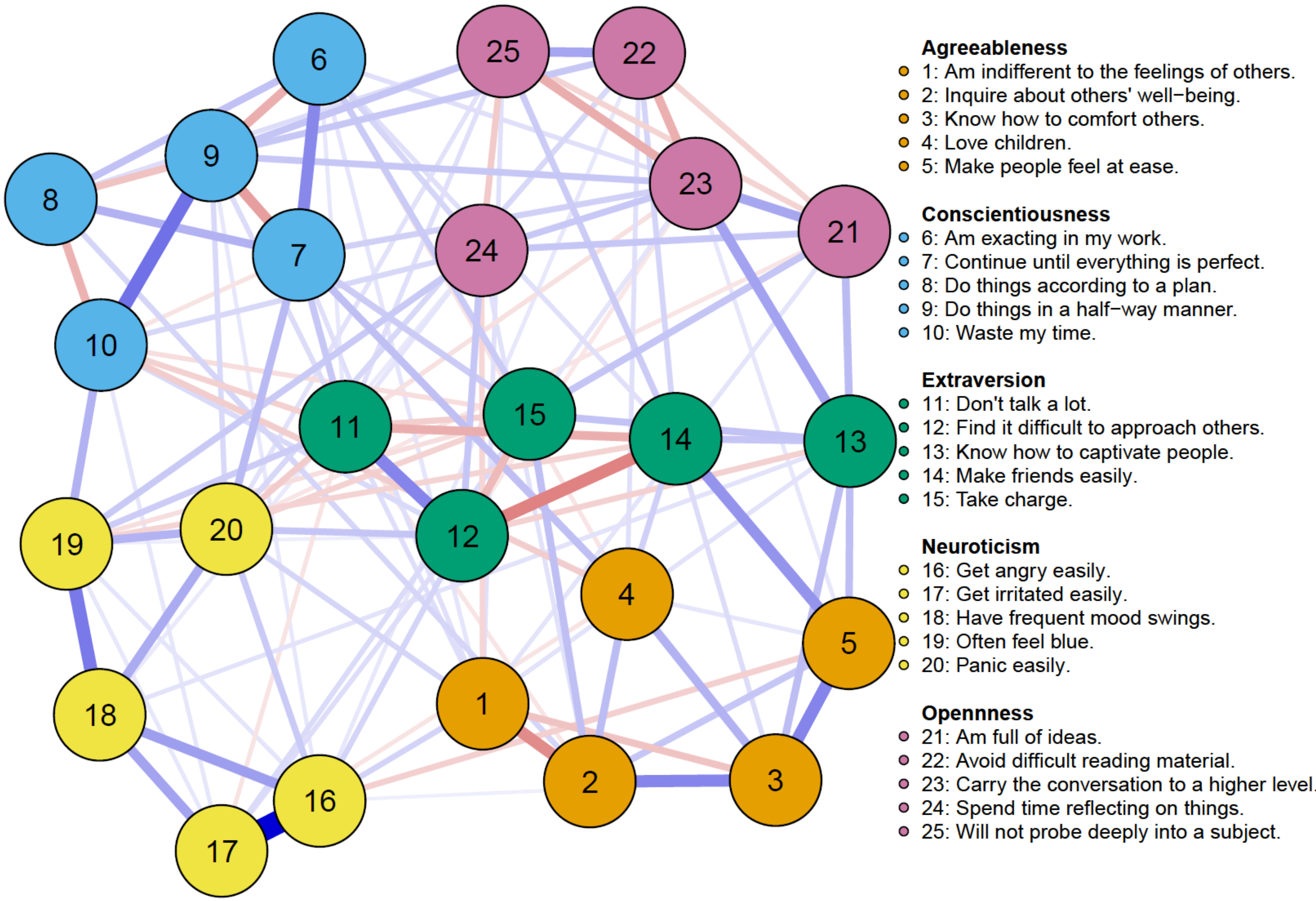
Part 4: Model search



Model Search

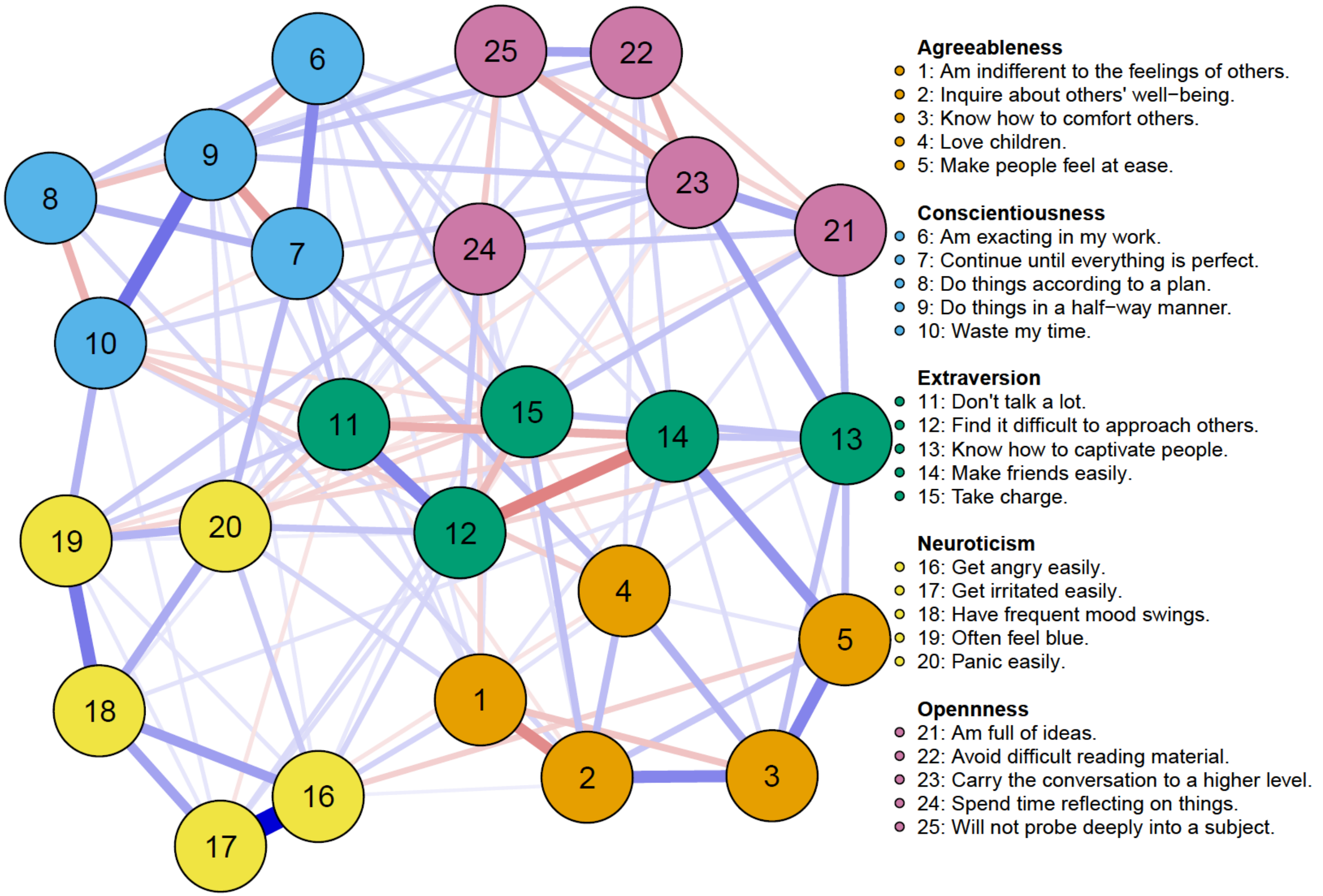
- Extensive model search strategies can be used to find the best model
- Uses unregularized (maximum likelihood) estimation with edges fixed to zero
- Simple stepwise model search algorithm:
  - stepup in (psychonetrics)
- More advanced stepwise model search algorithms:
  - modelsearch (psychonetrics)
  - ggmModSelect (qgraph)
- Can be **slow**

Step-up model search (psychonetrics)




```
mod <- ggm(bfiSub, omega = "empty") %>% runmodel %>%
       stepup

net <- getmatrix(mod, "omega")
```

Step-up model search (psychonetrics)

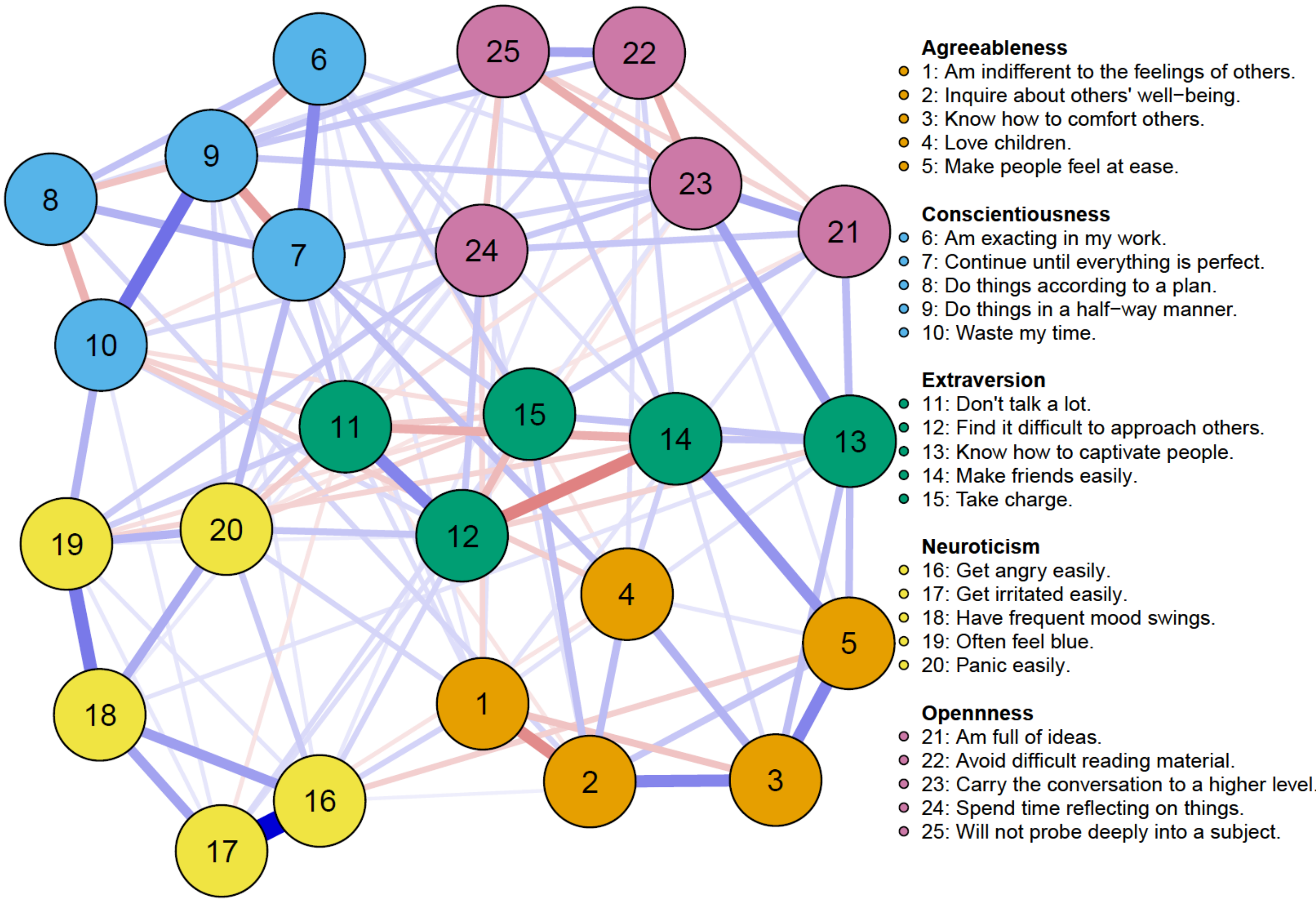



```
mod <- ggm(bfiSub, omega = "empty") %>% runmodel %>%
       stepup
```

Start with empty model

```
net <- getmatrix(mod, "omega")
```

Step-up model search (psychonetrics)

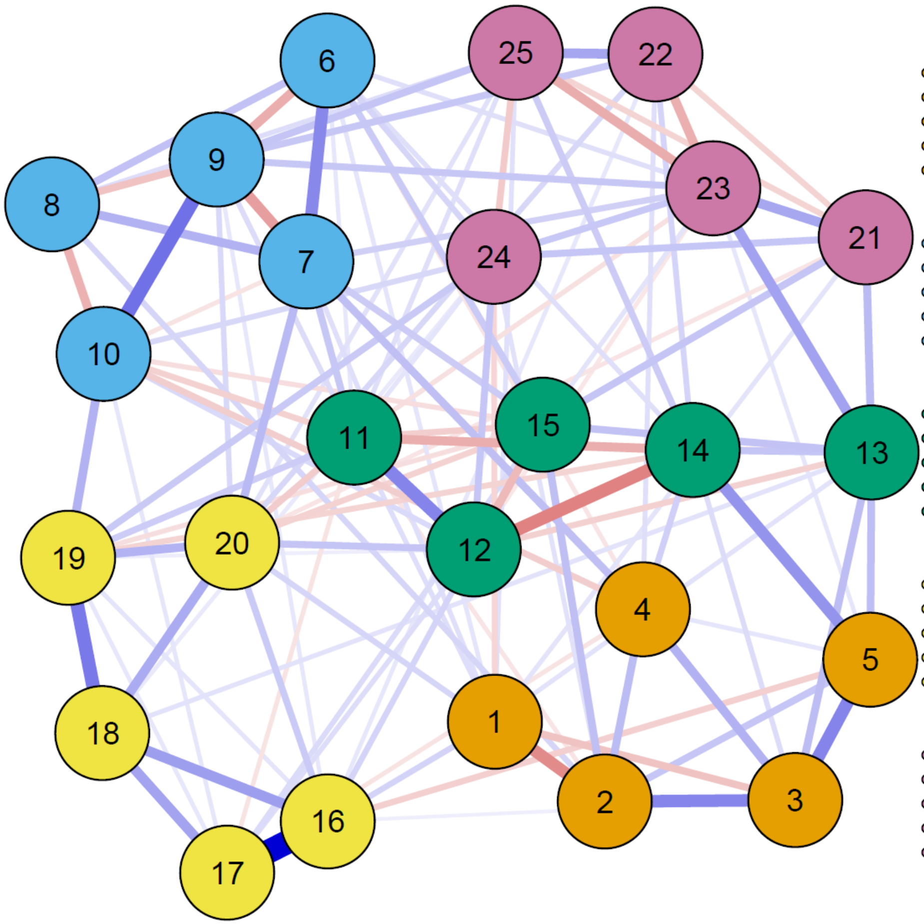



```
mod <- ggm(bfiSub, omega = "empty") %>% runmodel %>%
       stepup
```

Add edges with highest modification index until BIC is optimized

```
net <- getmatrix(mod, "omega")
```

Step-up model search (psychonetrics)



Agreeableness
- 1: Am indifferent to the feelings of others.
- 2: Inquire about others' well−being.
- 3: Know how to comfort others.
- 4: Love children.
- 5: Make people feel at ease.

Conscientiousness
- 6: Am exacting in my work.
- 7: Continue until everything is perfec
- 8: Do things according to a plan.
- 9: Do things in a half−way manner.
- 10: Waste my time.

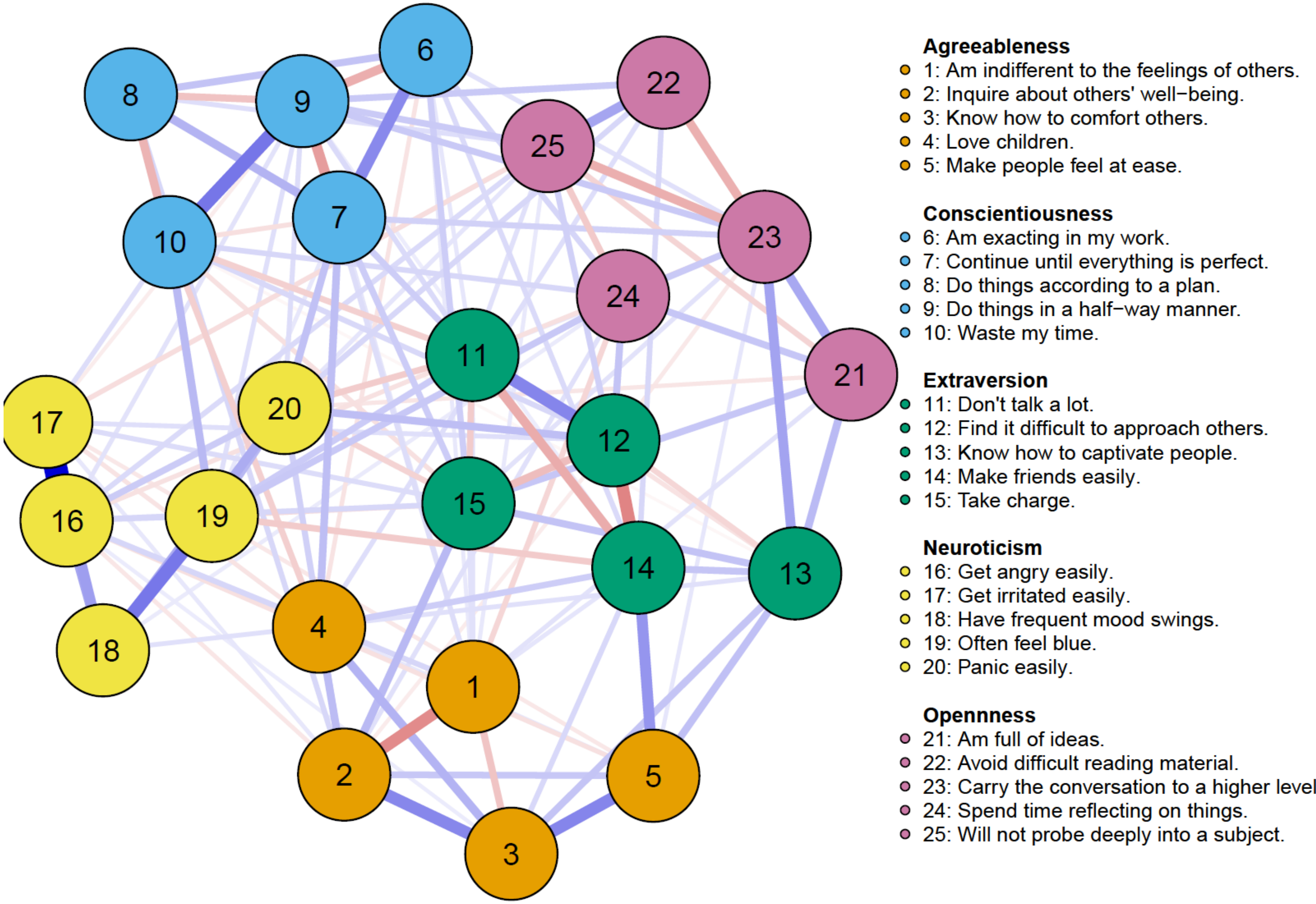
Extraversion
- 11: Don't talk a lot.
- 12: Find it difficult to approach others
- 13: Know how to captivate people.
- 14: Make friends easily.
- 15: Take charge.

Neuroticism
- 16: Get angry easily.
- 17: Get irritated easily.
- 18: Have frequent mood swings.
- 19: Often feel blue.
- 20: Panic easily.

Opennness
- 21: Am full of ideas.
- 22: Avoid difficult reading material.
- 23: Carry the conversation to a higher level.
- 24: Spend time reflecting on things.
- 25: Will not probe deeply into a subject.

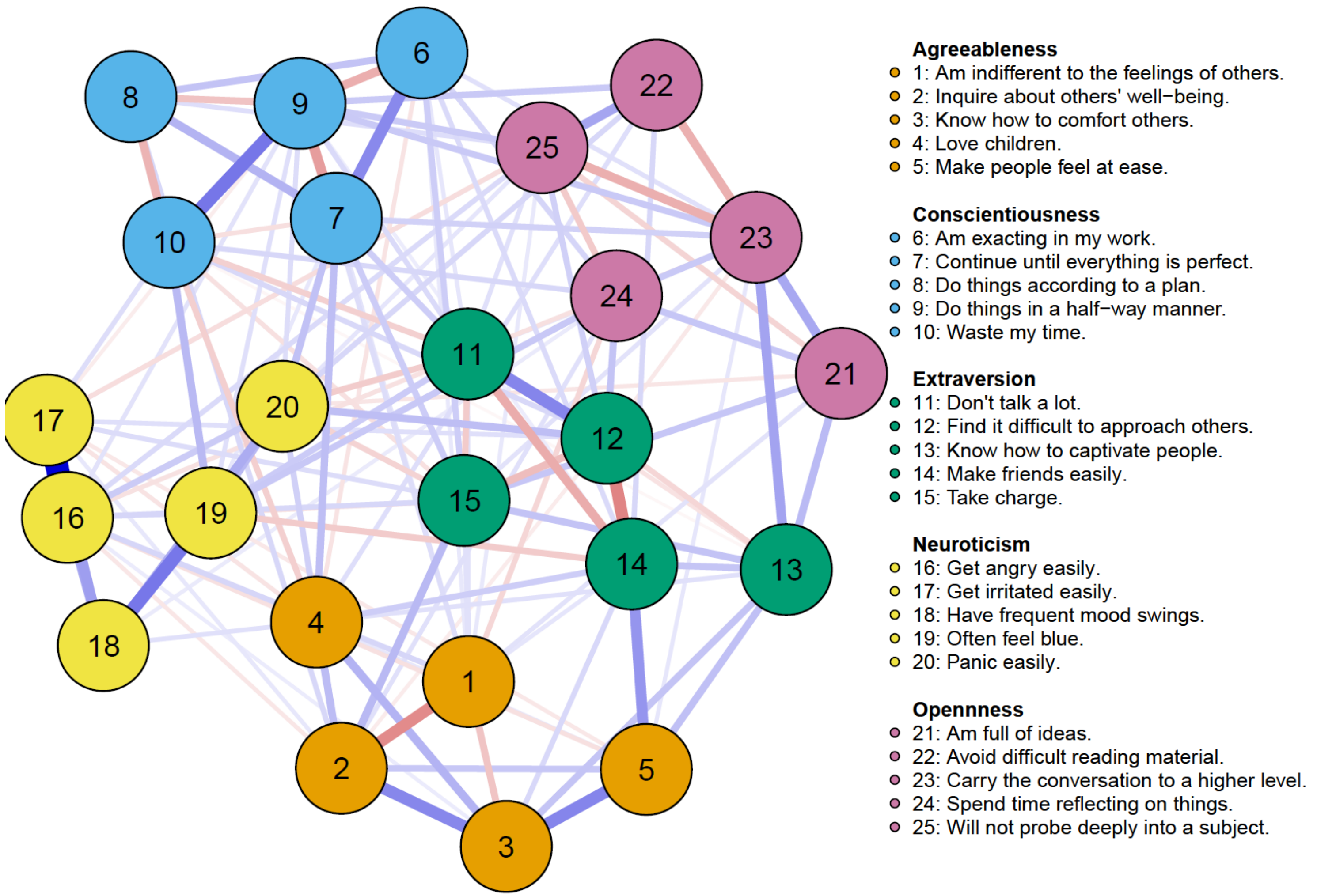
```
mod <- ggm(bfiSub, omega = "empty") %>% runmodel %>%
       stepup

net <- getmatrix(mod, "omega")
```

Prune -> step-up model search (psychonetrics)



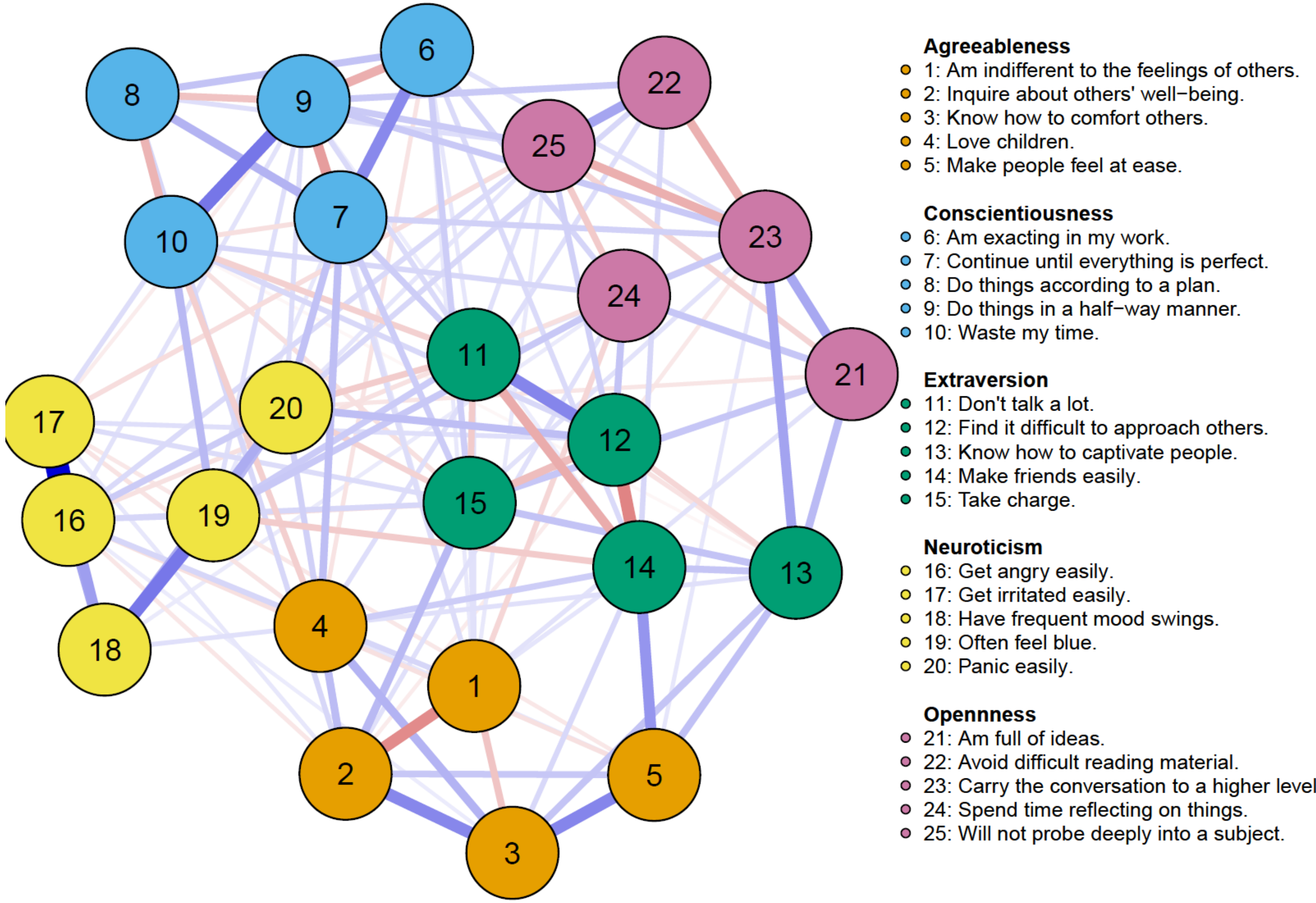

```
mod <- ggm(bfiSub) %>% runmodel %>%
       prune(alpha = 0.01) %>% stepup

net <- getmatrix(mod, "omega")
```

Prune -> step-up model search (psychonetrics)




```
mod <- ggm(bfiSub) %>% runmodel %>%
       prune(alpha = 0.01) %>% stepup

net <- getmatrix(mod, "omega")
```

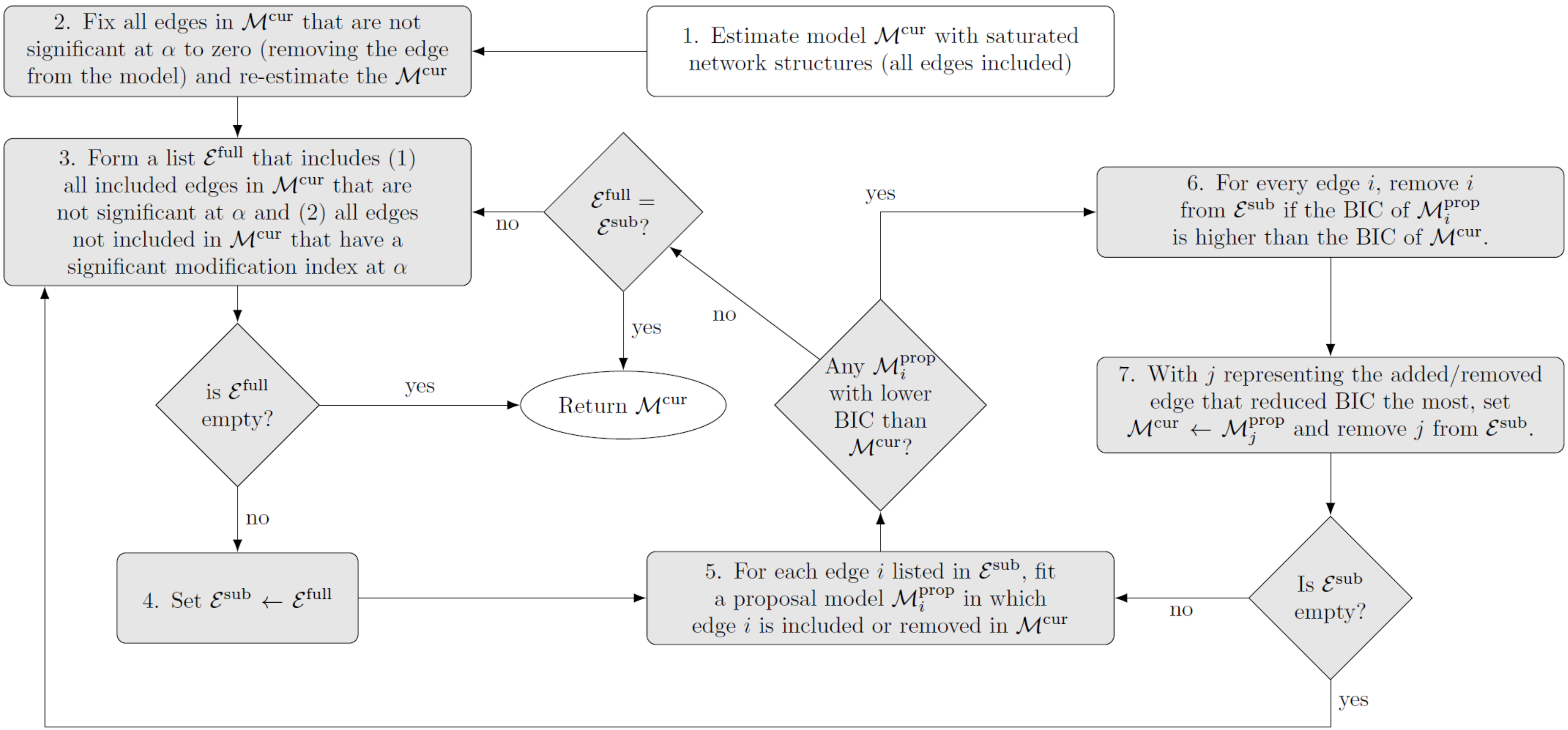
Prune -> step-up model search (psychonetrics)



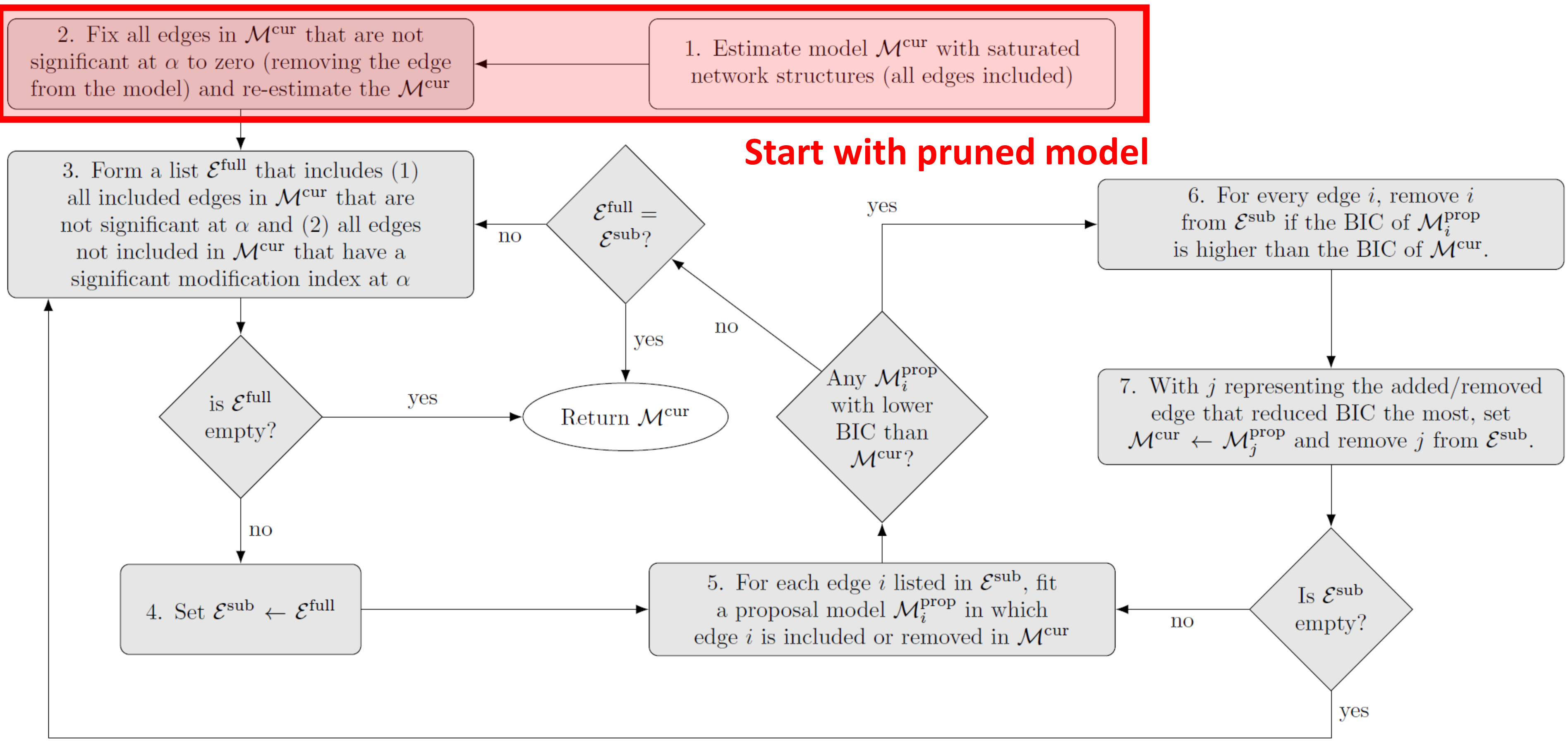

```
mod <- ggm(bfiSub) %>% runmodel %>%
         prune(alpha = 0.01) %>% stepup

net <- getmatrix(mod, "omega")
```

Longer combinations are possible!

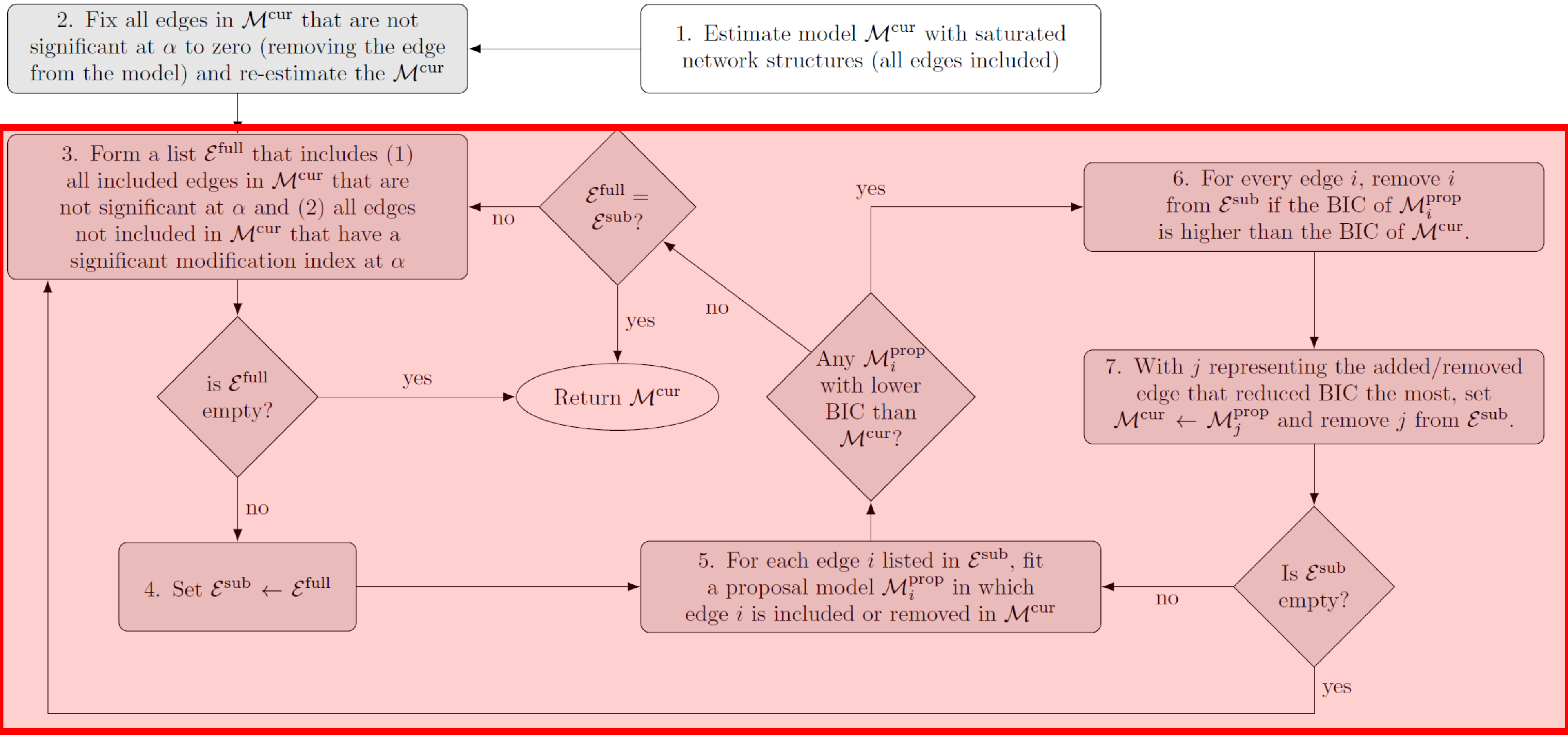
The modelsearch algorithm (psychonetrics)



The modelsearch algorithm (psychonetrics)

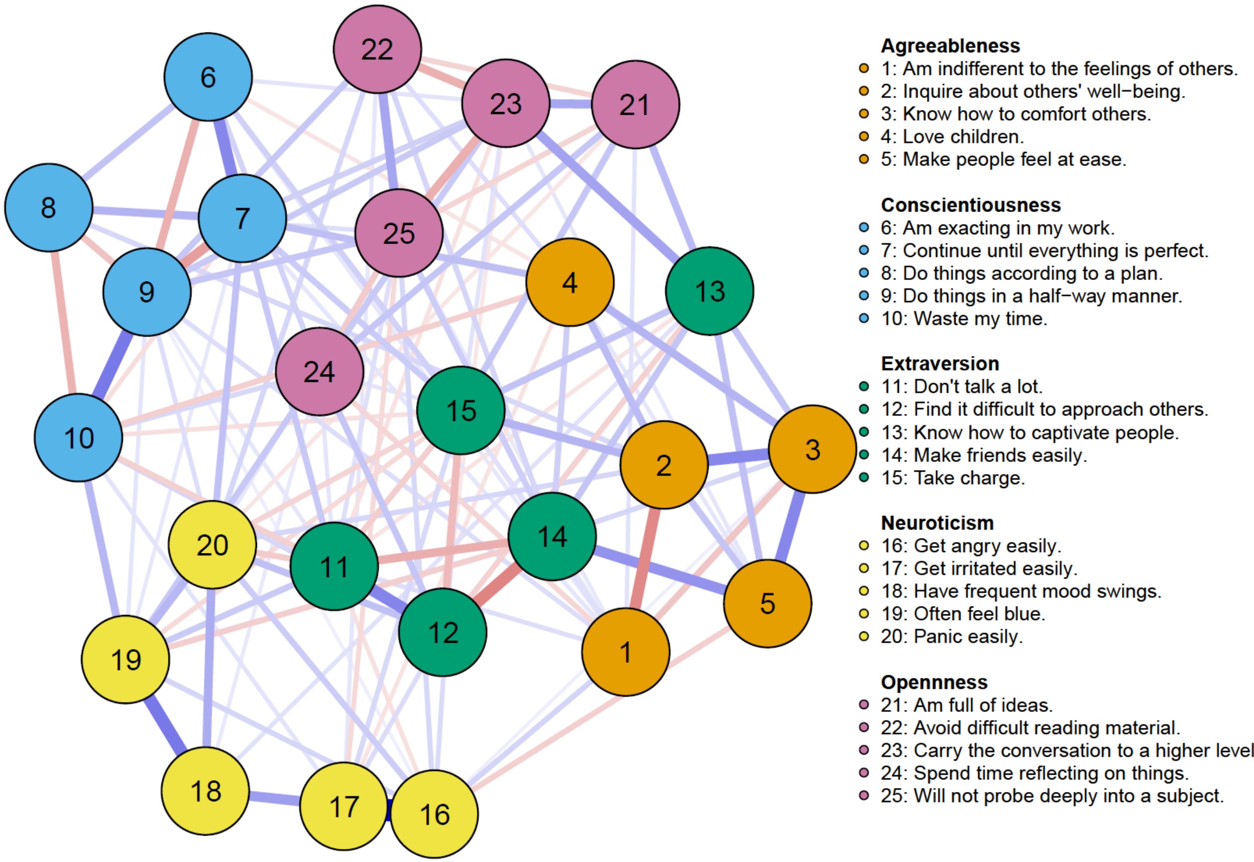


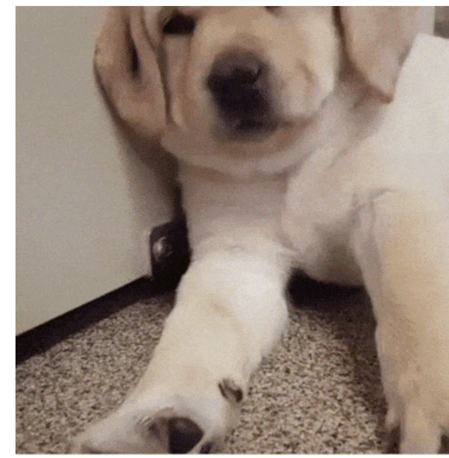
The modelsearch algorithm (psychonetrics)



Iteratively add/remove edges (using p-values / Mis) until BIC is optimized

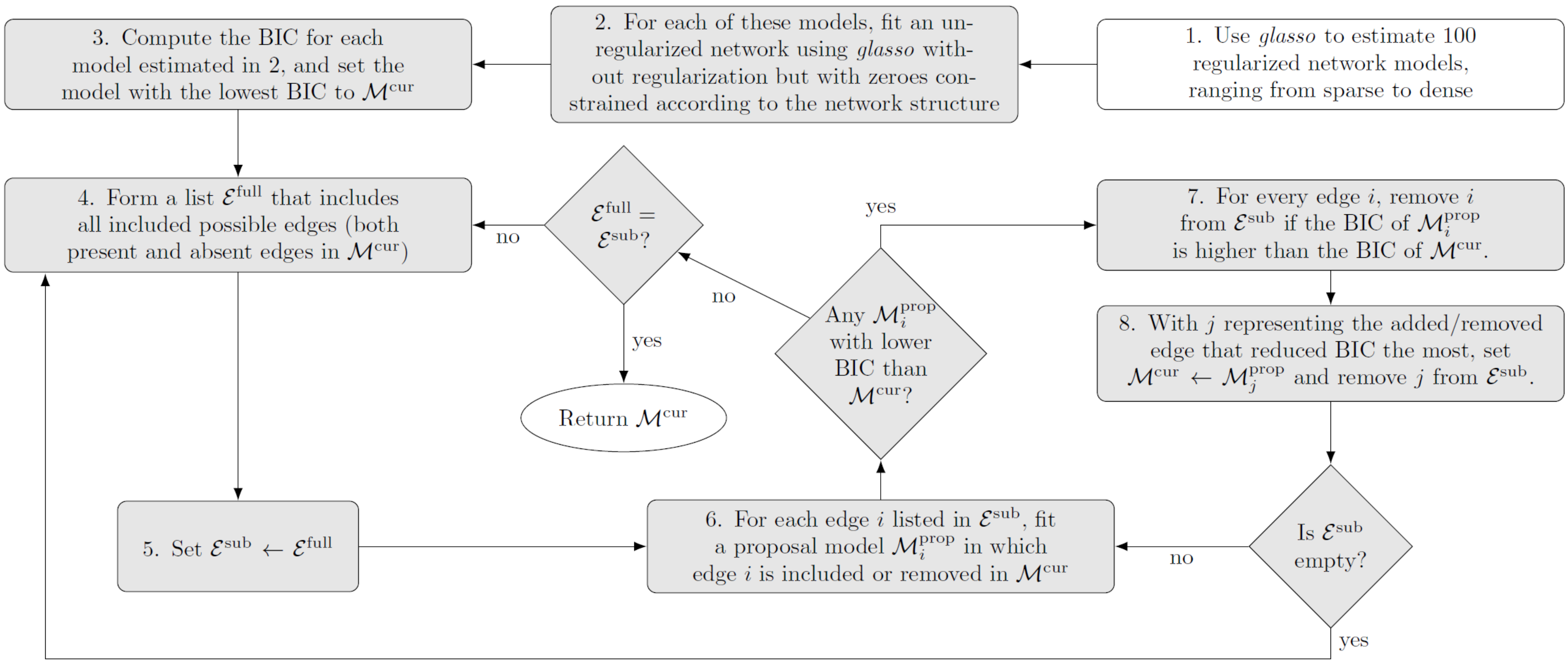
Model search (psychonetrics)





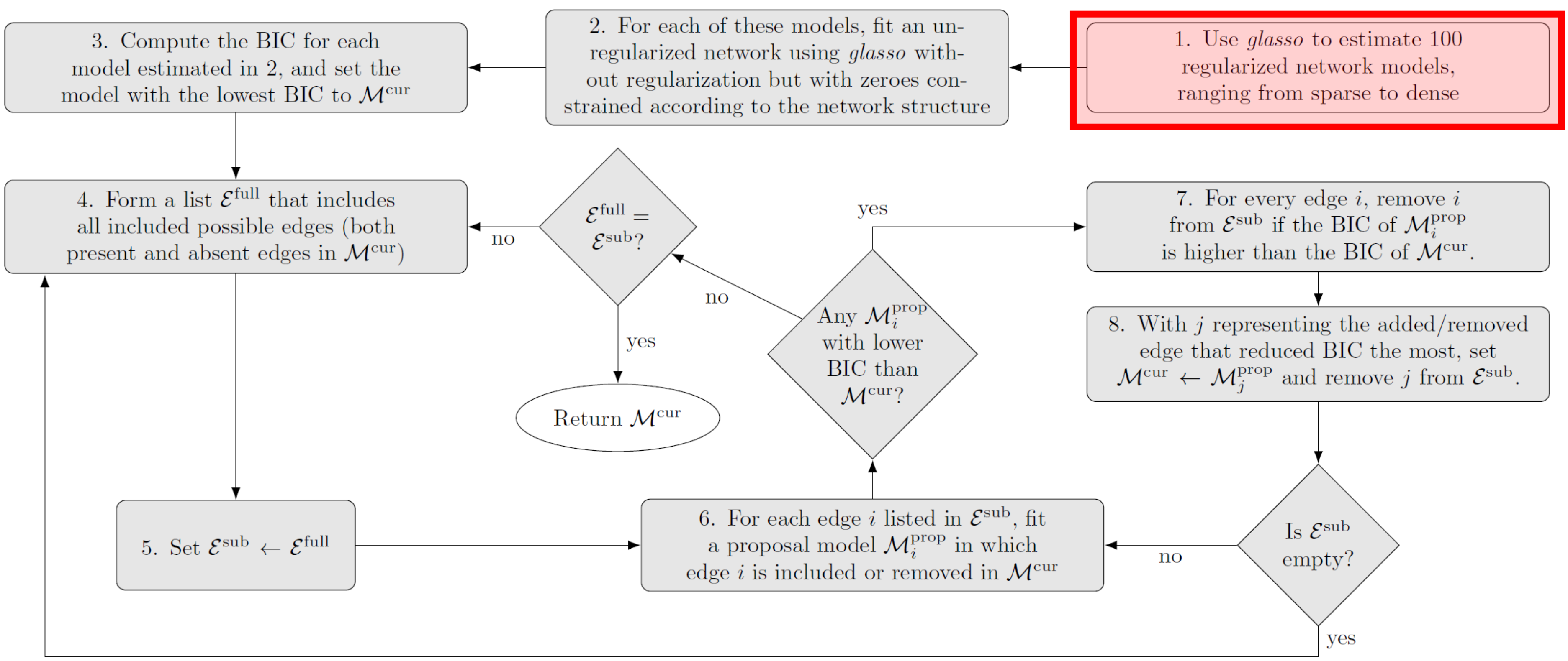
```
mod <- ggm(bfiSub) %>% runmodel %>% prune(alpha = 0.01) %>%
                modelsearch

net <- getmatrix(mod, "omega")
```

The ggmModSelect algorithm (qgraph)

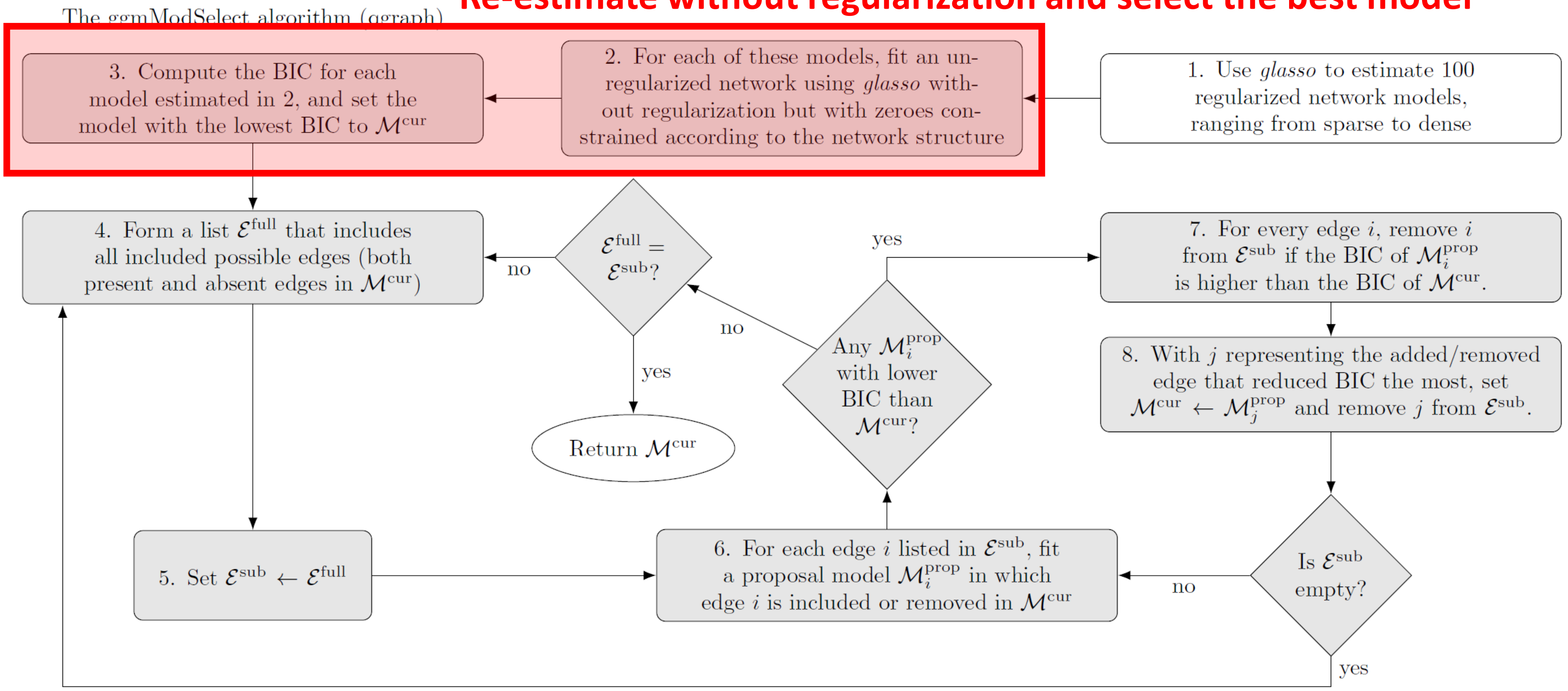


The ggmModSelect algorithm (qgraph)

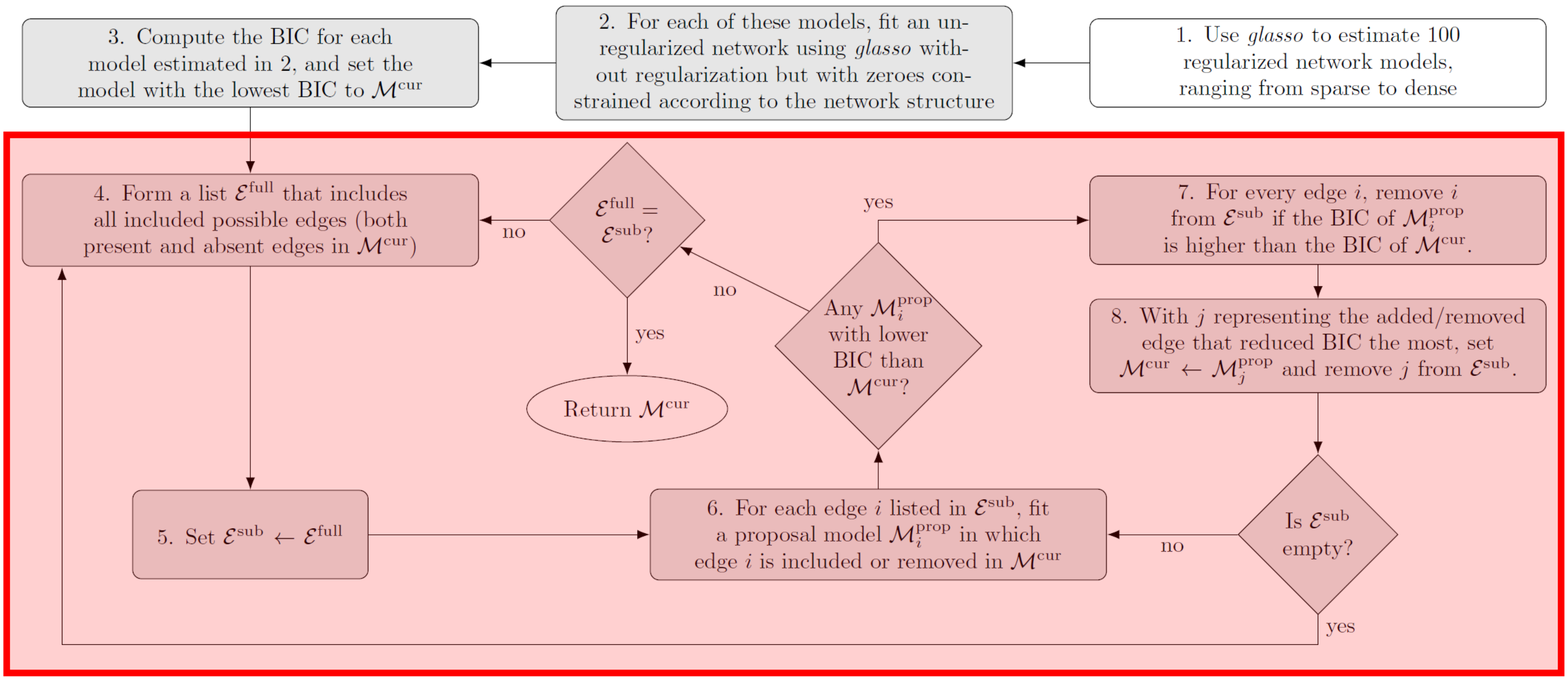
Start with glasso



Re-estimate without regularization and select the best model

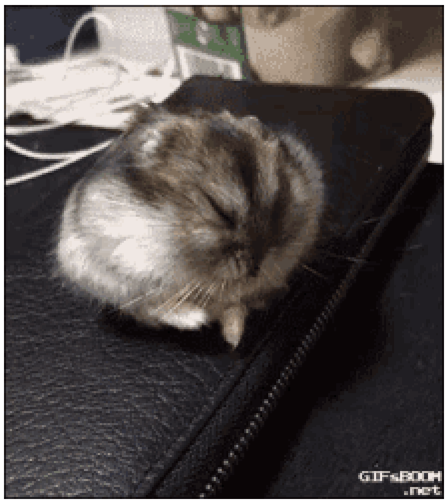


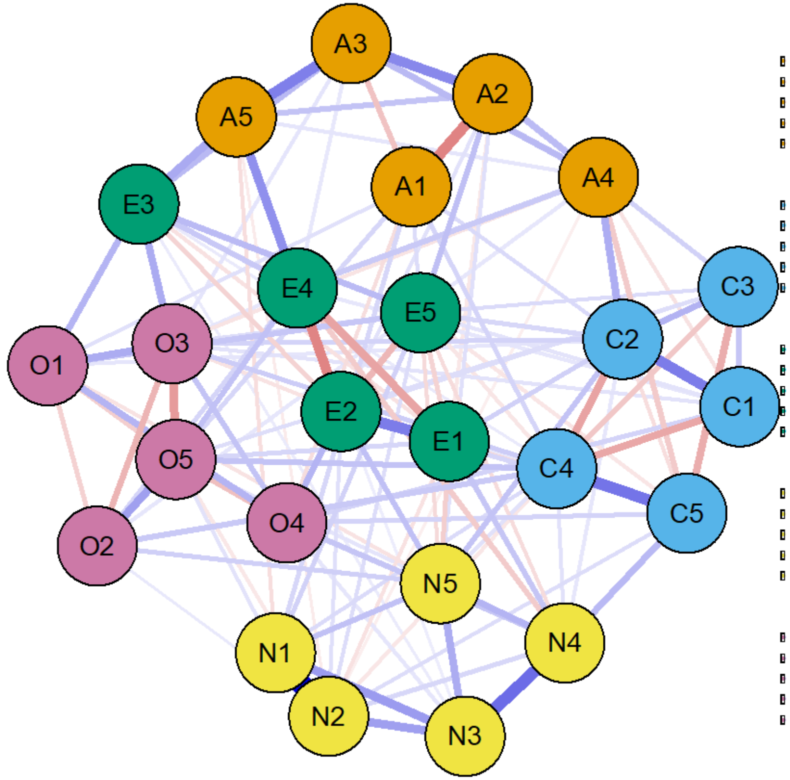
The ggmModSelect algorithm (qgraph)



If stepwise = TRUE: Iteratively add and remove edges until BIC is optimized

ggmModSelect with stepwise estimation (= slow!)





Agreeableness
- A1: Am indifferent to the feelings of others.
- A2: Inquire about others' well−being.
- A3: Know how to comfort others.
- A4: Love children.
- A5: Make people feel at ease.

Conscientiousness
- C1: Am exacting in my work.
- C2: Continue until everything is perfect.
- C3: Do things according to a plan.
- C4: Do things in a half−way manner.
- C5: Waste my time.

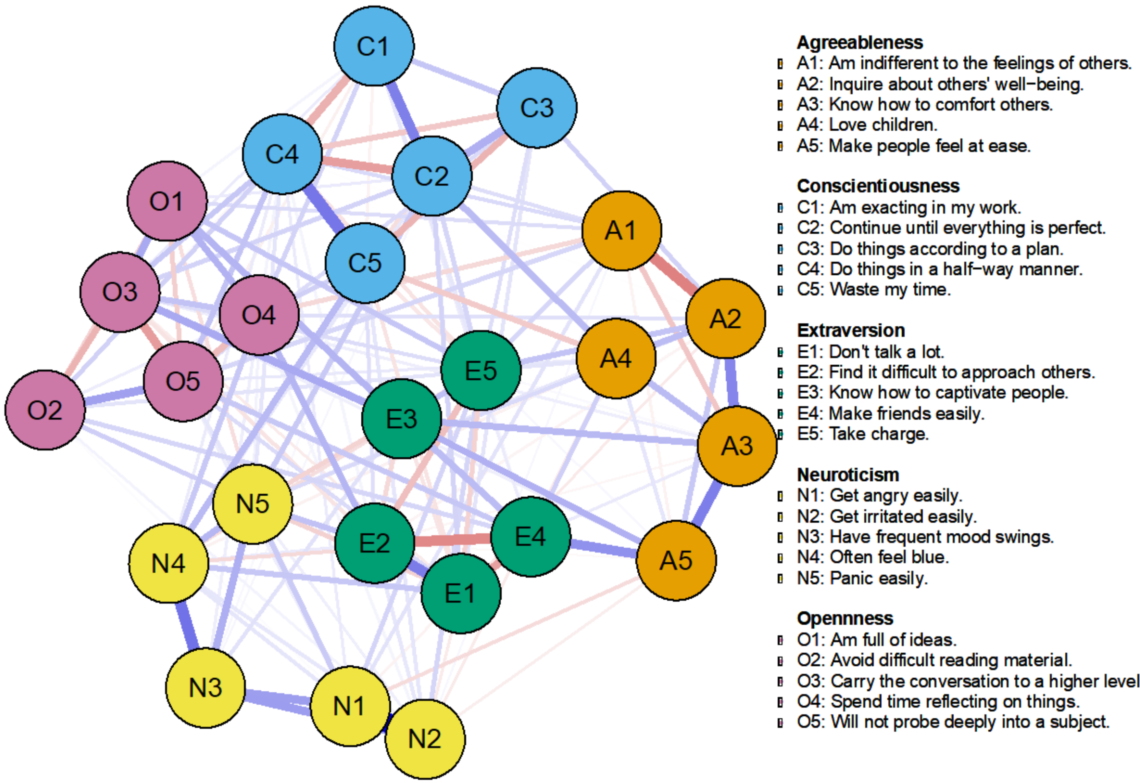
Extraversion
- E1: Don't talk a lot.
- E2: Find it difficult to approach others.
- E3: Know how to captivate people.
- E4: Make friends easily.
- E5: Take charge.

Neuroticism
- N1: Get angry easily.
- N2: Get irritated easily.
- N3: Have frequent mood swings.
- N4: Often feel blue.
- N5: Panic easily.

Openness
- O1: Am full of ideas.
- O2: Avoid difficult reading material.
- O3: Carry the conversation to a higher level.
- O4: Spend time reflecting on things.
- O5: Will not probe deeply into a subject.

```
net_ggmModSelect <- estimateNetwork(bfiSub,
     default = "ggmModSelect",
     corMethod = "spearman")
```

ggmModSelect with stepwise estimation (= fast!)




```
net_ggmModSelect2 <- estimateNetwork(bfiSub,
      stepwise = FALSE,
      default = "ggmModSelect",
      corMethod = "spearman")
```

Model search

+

- Good performance in sensitivity and specificity
  - Conservative but also powerful
- Can converge to the true model
- Non-biased estimates
- Local optimum

-

- Very slow with stepwise estimation
  - Especially for large networks
- No regularization
  - Potentially prominent false edges

Model Selection

Part 5: What method to use

Summary!

- 1. Thresholding / pruning: estimate saturated model, then remove edges based on some criterion (e.g., significance, Bayes Factor, false discovery rate)
  - Pros: very fast, non-biased estimates, can feature fixed false positive rate per edge
  - Cons: no model selection, not always possible, might not converge to true model, possibly more sampling variation at low sample sizes (especially in Ising model)
- 2. Regularization: technique for joint model selection and parameter estimation from machine learning
  - Pros: fast, can be used when normal methods fail (e.g., low sample size or violated assumptions), shrinkage to zero can improve visual representation
  - Cons: always biased estimates, poorer performance in high sample sizes
- 3. Model search: Search an optimal fitting model by adding and removing edges to optimize some criterion (e.g., stepwise search)
  - Pros: model selection, non-biased estimates, can converge to true model
  - Cons: can be slow (especially when performing multivariate model search), possibly more sampling variation at low sample sizes

Bayesian thresholding

```
select(explore(Data))
select(estimate(Data))
```

```
ggm(Data)

# In combination with:
runmodel()
prune()
stepup()
modelsearch()
```

Pruning / model search

An R package for confirmatory network modeling

```
# Estimate a network:
library("bootnet")
Network <- estimateNetwork(Data, default = "...")
```

Default	Model	Data	Estimation	Penalty	Selection	Package
pcor	GGM	Continuous, ordinal	Inverse	None	None, sig, FDR	qgraph
EBICglasso	GGM	Continuous, ordinal	Inverse	glasso	EBIC	qgraph
ggmModSelect	GGM	Continuous, ordinal	Stepwise model fit	None	BIC	qgraph
IsingFit	Ising	Binary	Nodewise	LASSO	EBIC	IsingFit
IsingSampler	Ising	Binary	Nodewise, MLE	None	None	IsingSampler
adalasso	GGM	Continuous	Nodewise	Adaptive LASSO	CV	parcor
huge	GGM	Continuous	Inverse	glasso	EBIC	huge
mgm	MGM	Continuous, count, binary, categorical	Nodewise	LASSO	EBIC, CV	mgm
relimp	"GGM"	Continuous	Nodewise	None	Any other default	relimpo

CV = Cross validation; Inverse = Inverse covariance matrix; MLE = Maximum (pseudo) likelihood

```
# Estimate a network:
library("bootnet")
Network <- estimateNetwork(Data, default = "...")
```

Thresholding (with arguments / bootstraps)

Default	Model	Data	Estimation	Penalty	Selection	Package
pcor	GGM	Continuous, ordinal	Inverse	None	None, sig, FDR	qgraph
EBICglasso	GGM	Continuous, ordinal	Inverse	glasso	EBIC	qgraph
ggmModSelect	GGM	Continuous, ordinal	Stepwise model fit	None	BIC	qgraph
IsingFit	Ising	Binary	Nodewise	LASSO	EBIC	IsingFit
IsingSampler	Ising	Binary	Nodewise, MLE	None	None	IsingSampler
adalasso	GGM	Continuous	Nodewise	Adaptive LASSO	CV	parcor
huge	GGM	Continuous	Inverse	glasso	EBIC	huge
mgm	MGM	Continuous, count, binary, categorical	Nodewise	LASSO	EBIC, CV	mgm
relimp	"GGM"	Continuous	Nodewise	None	Any other default	relimpo

CV = Cross validation; Inverse = Inverse covariance matrix; MLE = Maximum (pseudo) likelihood

```
# Estimate a network:
library("bootnet")
Network <- estimateNetwork(Data, default = "...")
```

Regularization

Default	Model	Data	Estimation	Penalty	Selection	Package
pcor	GGM	Continuous, ordinal	Inverse	None	None, sig, FDR	qgraph
EBICglasso	GGM	Continuous, ordinal	Inverse	glasso	EBIC	qgraph
ggmModSelect	GGM	Continuous, ordinal	Stepwise model fit	None	BIC	qgraph
IsingFit	Ising	Binary	Nodewise	LASSO	EBIC	IsingFit
IsingSampler	Ising	Binary	Nodewise, MLE	None	None	IsingSampler
adalasso	GGM	Continuous	Nodewise	Adaptive LASSO	CV	parcor
huge	GGM	Continuous	Inverse	glasso	EBIC	huge
mgm	MGM	Continuous, count, binary, categorical	Nodewise	LASSO	EBIC, CV	mgm
relimp	"GGM"	Continuous	Nodewise	None	Any other default	relimpo

CV = Cross validation; Inverse = Inverse covariance matrix; MLE = Maximum (pseudo) likelihood

```
# Estimate a network:
library("bootnet")
Network <- estimateNetwork(Data, default = "...")
```

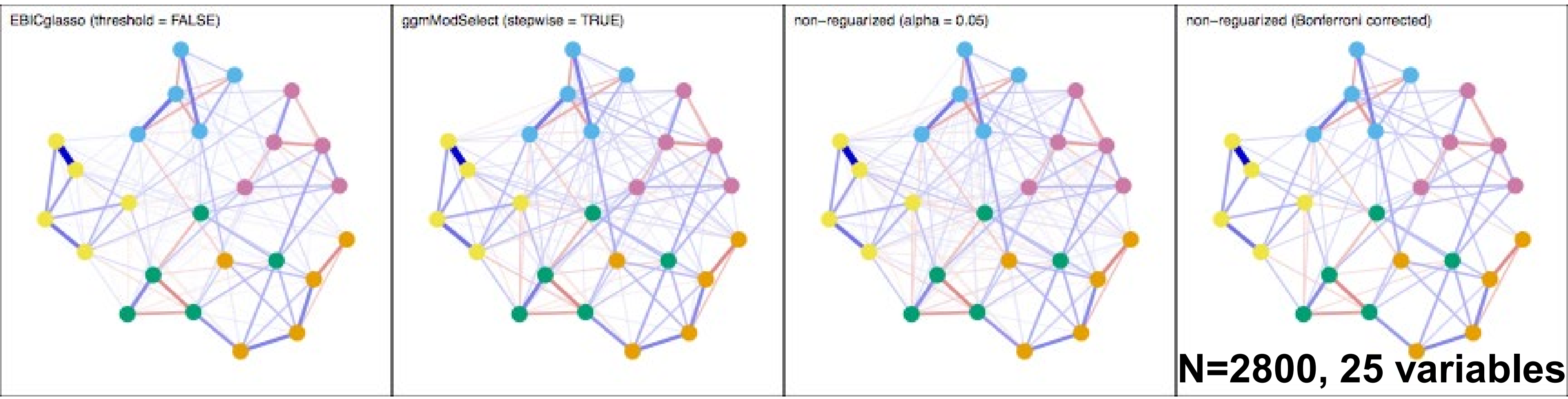
Model search

Default	Model	Data	Estimation	Penalty	Selection	Package
pcor	GGM	Continuous, ordinal	Inverse	None	None, sig, FDR	qgraph
EBICglasso	GGM	Continuous, ordinal	Inverse	glasso	EBIC	qgraph
ggmModSelect	GGM	Continuous, ordinal	Stepwise model fit	None	BIC	qgraph
IsingFit	Ising	Binary	Nodewise	LASSO	EBIC	IsingFit
IsingSampler	Ising	Binary	Nodewise, MLE	None	None	IsingSampler
adalasso	GGM	Continuous	Nodewise	Adaptive LASSO	CV	parcor
huge	GGM	Continuous	Inverse	glasso	EBIC	huge
mgm	MGM	Continuous, count, binary, categorical	Nodewise	LASSO	EBIC, CV	mgm
relimp	"GGM"	Continuous	Nodewise	None	Any other default	relimpo

CV = Cross validation; Inverse = Inverse covariance matrix; MLE = Maximum (pseudo) likelihood

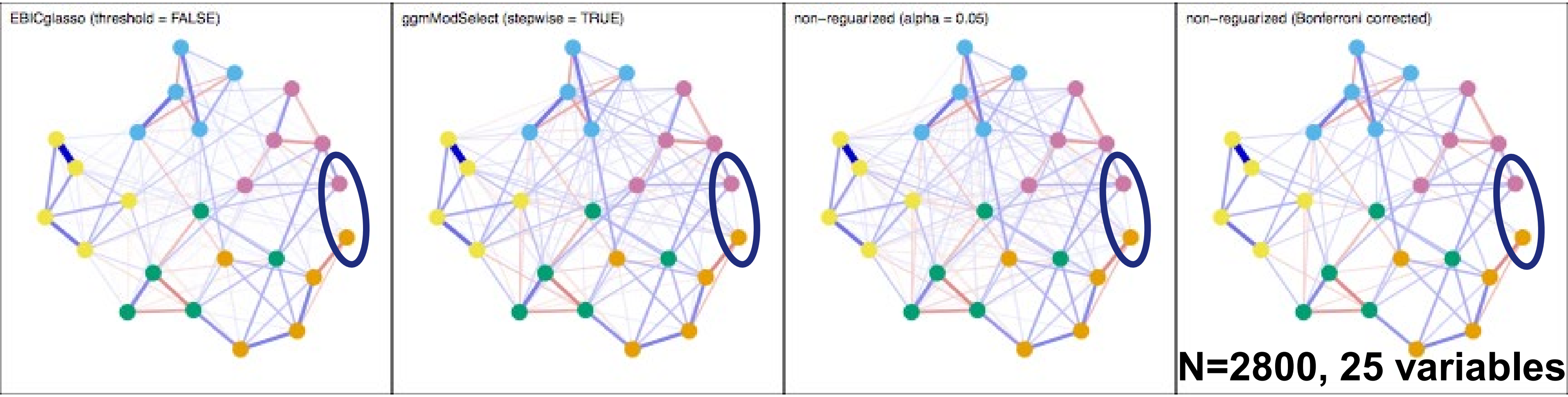
What method to use?

Regularization | Model selection | Pruning (sig) | Pruning (bonferroni)



What method to use?

Regularization | Model selection | Pruning (sig) | Pruning (bonferroni)



Psychological Medicine

cambridge.org/psm

Toward incorporating genetic risk scores into symptom networks of psychosis

Adela-Maria Isvoranu[1], Sinan Guloksuz[2,3], Sacha Epskamp[1], Jim van Os[4], Denny Borsboom[1] and GROUP Investigators†*

[1]Department of Psychology, Psychological Methods, University of Amsterdam, Amsterdam, The Netherlands; [2]Department of Psychiatry and Neuropsychology, School of Mental Health and Neuroscience, Maastricht University Medical Center, Maastricht, The Netherlands; [3]Department of Psychiatry, Yale School of Medicine, New Haven, CT, USA and [4]Utrecht University Medical Centre, Utrecht, The Netherlands

Original Article

Cite this article: Isvoranu A-M, Guloksuz S, Epskamp S, van Os J, Borsboom D, GROUP Investigators (2019). Toward incorporating genetic risk scores into symptom networks of psychosis. *Psychological Medicine* 1–8. https://doi.org/10.1017/S003329171900045X

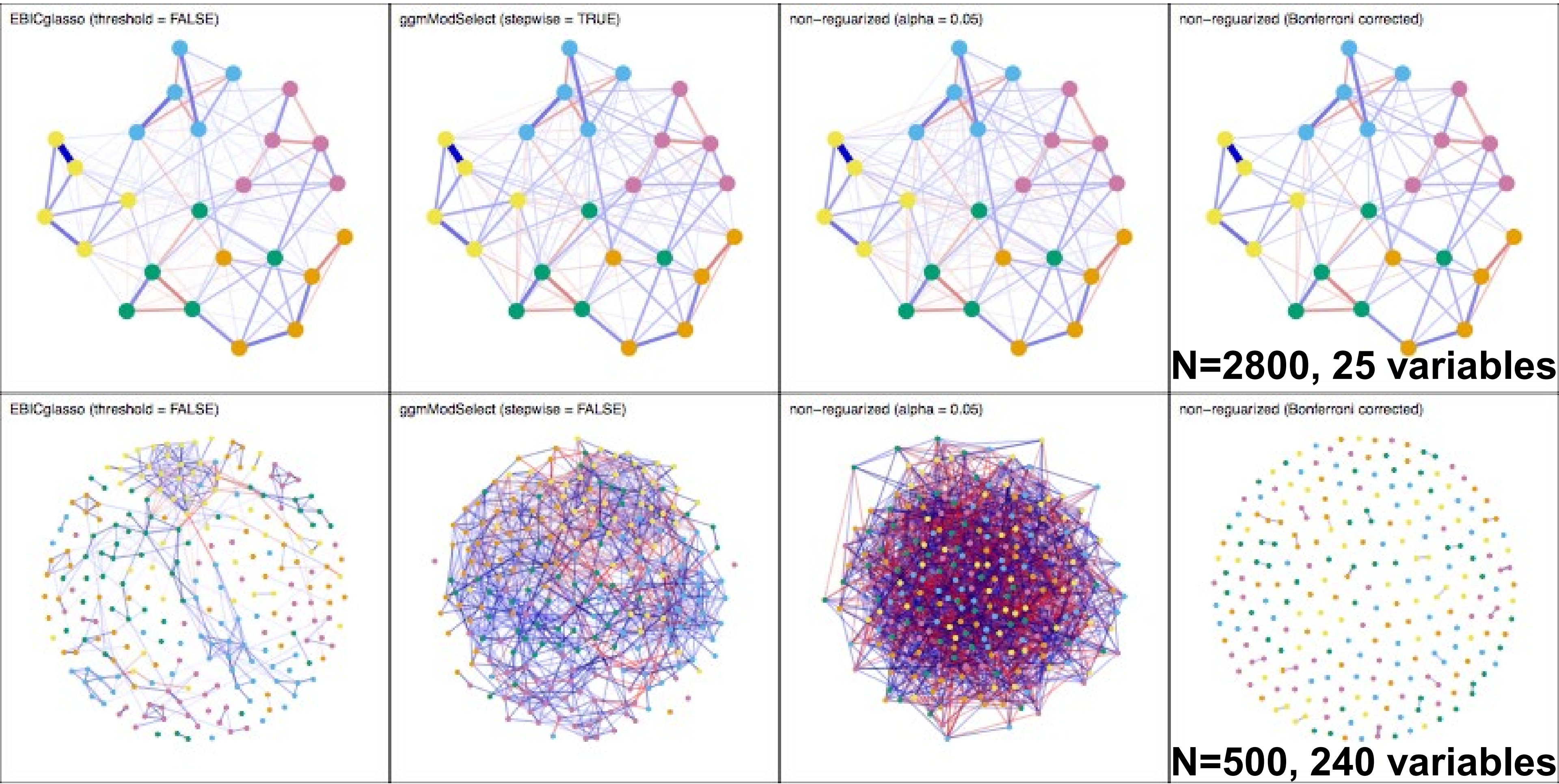
Received: 7 October 2018
Revised: 14 February 2019

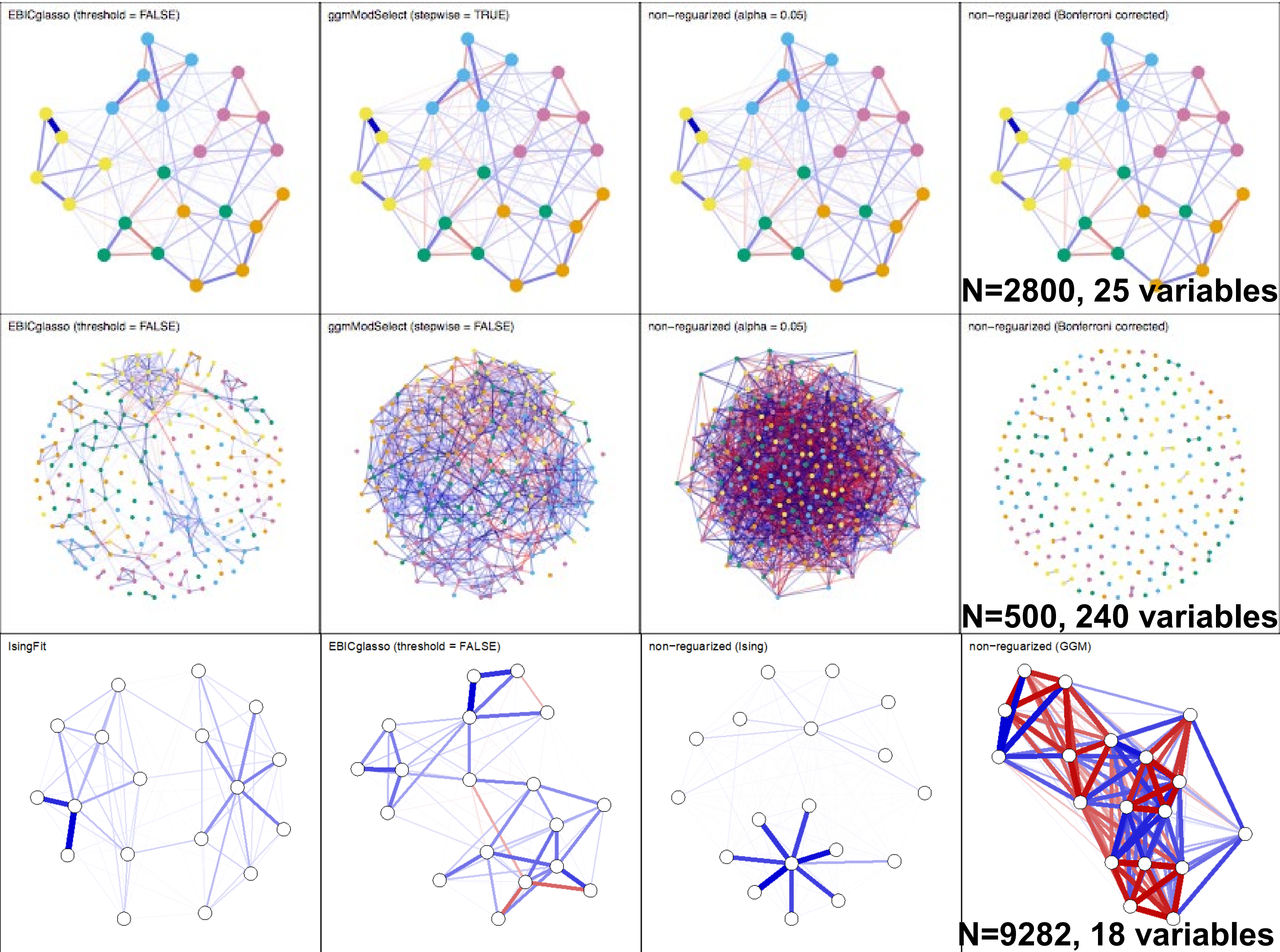
Abstract

Background. Psychosis spectrum disorder is a heterogeneous, multifactorial clinical phenotype, known to have a high heritability, only a minor portion of which can be explained by

What method to use?

Regularization | Model selection | Pruning (sig) | Pruning (bonferroni)




EBICglasso (threshold = FALSE)
ggmModSelect (stepwise = TRUE)
non−reguarized (alpha = 0.05)
non−reguarized (Bonferroni corrected)
N=2800, 25 variables
EBICglasso (threshold = FALSE)
ggmModSelect (stepwise = FALSE)
non−reguarized (alpha = 0.05)
non−reguarized (Bonferroni corrected)
N=500, 240 variables
IsingFit
EBICglasso (threshold = FALSE)
non−reguarized (Ising)
non−reguarized (GGM)
N=9282, 18 variables

```
# Estimate a network:
library("bootnet")
Network <- estimateNetwork(Data, default = "...")
```

Default	Model	Data	Estimation	Penalty	Selection	Package
pcor	GGM	Continuous, ordinal	Inverse	None	None, sig, FDR	qgraph
EBICglasso	GGM	Continuous, ordinal	Inverse	glasso	EBIC	qgraph
ggmModSelect	GGM	Continuous, ordinal	Stepwise model fit	None	BIC	qgraph
IsingFit	Ising	Binary	Nodewise	LASSO	EBIC	IsingFit
IsingSampler	Ising	Binary	Nodewise, MLE	None	None	IsingSampler
adalasso	GGM	Continuous	Nodewise	Adaptive LASSO	CV	parcor
huge	GGM	Continuous	Inverse	glasso	EBIC	huge
mgm	MGM	Continuous, count, binary, categorical	Nodewise	LASSO	EBIC, CV	mgm
relimp	"GGM"	Continuous	Nodewise	None	Any other default	relimpo

CV = Cross validation; Inverse = Inverse covariance matrix; MLE = Maximum (pseudo) likelihood

```
# Estimate a network:
library("bootnet")
Network <- estimateNetwork(Data, default = "...")
```

Gaussian data (high *n*)

Default	Model	Data	Estimation	Penalty	Selection	Package
pcor	GGM	Continuous, ordinal	Inverse	None	None, sig, FDR	qgraph
EBICglasso	GGM	Continuous, ordinal	Inverse	glasso	EBIC	qgraph
ggmModSelect	GGM	Continuous, ordinal	Stepwise model fit	None	BIC	qgraph
IsingFit	Ising	Binary	Nodewise	LASSO	EBIC	IsingFit
IsingSampler	Ising	Binary	Nodewise, MLE	None	None	IsingSampler
adalasso	GGM	Continuous	Nodewise	Adaptive LASSO	CV	parcor
huge	GGM	Continuous	Inverse	glasso	EBIC	huge
mgm	MGM	Continuous, count, binary, categorical	Nodewise	LASSO	EBIC, CV	mgm
relimp	"GGM"	Continuous	Nodewise	None	Any other default	relimpo

CV = Cross validation; Inverse = Inverse covariance matrix; MLE = Maximum (pseudo) likelihood

```
# Estimate a network:
library("bootnet")
Network <- estimateNetwork(Data, default = "...")
```

Gaussian data (low *n*)

Default	Model	Data	Estimation	Penalty	Selection	Package
pcor	GGM	Continuous, ordinal	Inverse	None	None, sig, FDR	qgraph
EBICglasso	GGM	Continuous, ordinal	Inverse	glasso	EBIC	qgraph
ggmModSelect	GGM	Continuous, ordinal	Stepwise model fit	None	BIC	qgraph
IsingFit	Ising	Binary	Nodewise	LASSO	EBIC	IsingFit
IsingSampler	Ising	Binary	Nodewise, MLE	None	None	IsingSampler
adalasso	GGM	Continuous	Nodewise	Adaptive LASSO	CV	parcor
huge	GGM	Continuous	Inverse	glasso	EBIC	huge
mgm	MGM	Continuous, count, binary, categorical	Nodewise	LASSO	EBIC, CV	mgm
relimp	"GGM"	Continuous	Nodewise	None	Any other default	relimpo

CV = Cross validation; Inverse = Inverse covariance matrix; MLE = Maximum (pseudo) likelihood

```
# Estimate a network:
library("bootnet")
Network <- estimateNetwork(Data, default = "...")
```

Binary data

Default	Model	Data	Estimation	Penalty	Selection	Package
pcor	GGM	Continuous, ordinal	Inverse	None	None, sig, FDR	qgraph
EBICglasso	GGM	Continuous, ordinal	Inverse	glasso	EBIC	qgraph
ggmModSelect	GGM	Continuous, ordinal	Stepwise model fit	None	BIC	qgraph
IsingFit	Ising	Binary	Nodewise	LASSO	EBIC	IsingFit
IsingSampler	Ising	Binary	Nodewise, MLE	None	None	IsingSampler
adalasso	GGM	Continuous	Nodewise	Adaptive LASSO	CV	parcor
huge	GGM	Continuous	Inverse	glasso	EBIC	huge
mgm	MGM	Continuous, count, binary, categorical	Nodewise	LASSO	EBIC, CV	mgm
relimp	"GGM"	Continuous	Nodewise	None	Any other default	relimpo

CV = Cross validation; Inverse = Inverse covariance matrix; MLE = Maximum (pseudo) likelihood

```
# Estimate a network:
library("bootnet")
Network <- estimateNetwork(Data, default = "...")
```

Mixed data / categorical data

Default	Model	Data	Estimation	Penalty	Selection	Package
pcor	GGM	Continuous, ordinal	Inverse	None	None, sig, FDR	qgraph
EBICglasso	GGM	Continuous, ordinal	Inverse	glasso	EBIC	qgraph
ggmModSelect	GGM	Continuous, ordinal	Stepwise model fit	None	BIC	qgraph
IsingFit	Ising	Binary	Nodewise	LASSO	EBIC	IsingFit
IsingSampler	Ising	Binary	Nodewise, MLE	None	None	IsingSampler
adalasso	GGM	Continuous	Nodewise	Adaptive LASSO	CV	parcor
huge	GGM	Continuous	Inverse	glasso	EBIC	huge
mgm	MGM	Continuous, count, binary, categorical	Nodewise	LASSO	EBIC, CV	mgm
relimp	"GGM"	Continuous	Nodewise	None	Any other default	relimpo

CV = Cross validation; Inverse = Inverse covariance matrix; MLE = Maximum (pseudo) likelihood

Model Selection

Part 6: Do it yourself simulation study

Do it yourself simulation studies

- Simulation studies can be very powerful to look at the performance of network estimation tools yourself
- Can also be used to determine sample size needed!
- Some measures we often look at:
  - Sensitivity: proportion of true edges included in estimated network
  - Specificity: proportion of true absent edges also not included in estimated network (1 – false positive rate)
  - Correlation: Pearson correlation between true and estimated edge weights
- The `netSimulator` function in *bootnet* makes this easy
- For more advanced simulation studies, see *parSim*
  - http://psychonetrics.org/2019/09/01/simulation-studies-in-r-with-the-parsim-package/

Simulation studies with netSimulator

1. We need a network to simulate under. Let's take the BFI network estimated with ggmModSelect:

```
truenet <- estimateNetwork(bfiSub,
      default = "ggmModSelect",
      stepwise = FALSE)
```

```
genGGM(...)
```

2. Simulate using netSimulator:

```
sims_EBICglasso <- netSimulator(truenet$graph,
    nCases = c(100,250,500,1000),
    nReps = 100,
    nCores = 8,
    default = "EBICglasso")
```

Simulation studies with netSimulator

1. We need a network to simulate under. Let's take the BFI network estimated with ggmModSelect:

```
truenet <- estimateNetwork(bfiSub,
      default = "ggmModSelect",
      stepwise = FALSE)
```

2. Simulate using netSimulator:

```
sims_EBICglasso <- netSimulator(truenet$graph,
    nCases = c(100,250,500,1000),
    nReps = 100,
    nCores = 8,
    default = "EBICglasso")
```

Sample size

Simulation studies with netSimulator

1. We need a network to simulate under. Let's take the BFI network estimated with ggmModSelect:

```
truenet <- estimateNetwork(bfiSub,
       default = "ggmModSelect",
       stepwise = FALSE)
```

2. Simulate using netSimulator:

```
sims_EBICglasso <- netSimulator(truenet$graph,
    nCases = c(100,250,500,1000),
    nReps = 100,
    nCores = 8,
    default = "EBICglasso")
```

Number of repetitions

Simulation studies with netSimulator

1. We need a network to simulate under. Let's take the BFI network estimated with ggmModSelect:

```
truenet <- estimateNetwork(bfiSub,
       default = "ggmModSelect",
       stepwise = FALSE)
```

2. Simulate using netSimulator:

```
sims_EBICglasso <- netSimulator(truenet$graph,
    nCases = c(100,250,500,1000),
    nReps = 100,
    nCores = 8,
    default = "EBICglasso")
```

Number of computer threads to use

Simulation studies with netSimulator

1. We need a network to simulate under. Let's take the BFI network estimated with ggmModSelect:

```
truenet <- estimateNetwork(bfiSub,
       default = "ggmModSelect",
       stepwise = FALSE)
```

2. Simulate using netSimulator:

```
sims_EBICglasso <- netSimulator(truenet$graph,
    nCases = c(100,250,500,1000),
    nReps = 100,
    nCores = 8,
    default = "EBICglasso")
```

Any argument to estimateNetwork(...)

Simulation studies with netSimulator

1. We need a network to simulate under. Let's take the BFI network estimated with ggmModSelect:

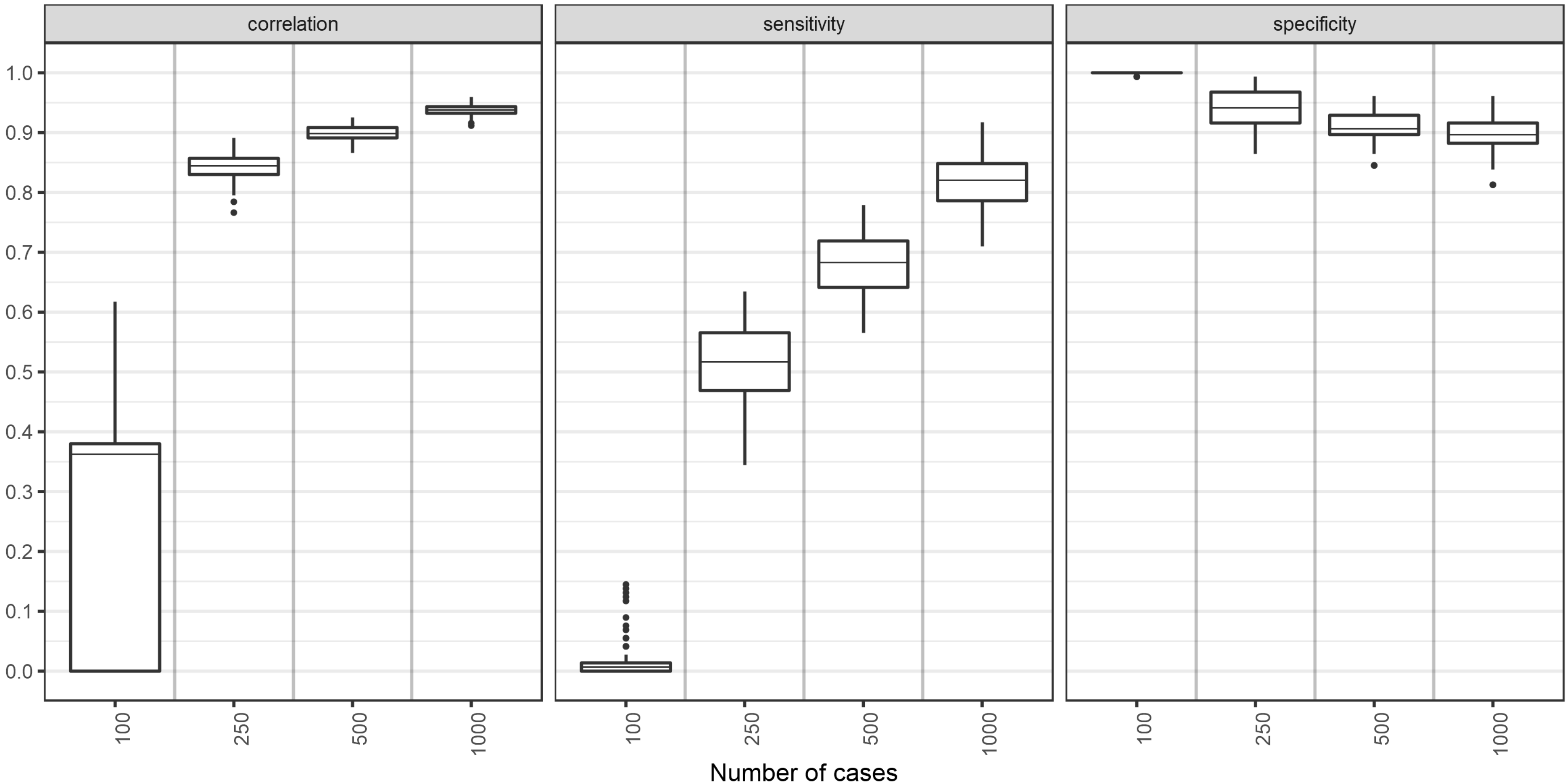
```
truenet <- estimateNetwork(bfiSub,
        default = "ggmModSelect",
        stepwise = FALSE)
```

2. Simulate using netSimulator:

```
sims_EBICglasso <- netSimulator(truenet$graph,
    nCases = c(100,250,500,1000),
    nReps = 100,
    nCores = 8,
    default = "EBICglasso")
```

3. Plot results:

```
plot(sims)
```



correlation
sensitivity
specificity
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
100
250
500
1000
Number of cases