IMPERIAL

Exposomics Analytics - Practical 4 Structural Equation Modelling

Contents

Overview of the Practical 2

Causal Effect Estimation 3

- Confounder, Mediator, Collider . . . 3
- Back-door criterion . . . 9
- Front-door criterion . . . 13

Sufficient Adjustment Set Determination 16

High-dimensional Mediation Analysis 17

Overview of the Practical

In this practical, you will engage with key concepts of causal inference and structural equation modelling (SEM), tools essential for disentangling complex relationships in epidemiological research. The primary aim of structural causal models is to investigate the total, direct, and indirect effects of an exposure on disease outcomes, expanding beyond the capabilities of traditional regression models, which are confined to investigating conditional associations.

By using directed acyclic graphs (DAGs), we can represent causal hypotheses between variables, helping to identify which variables must be adjusted for in statistical models to ensure accurate causal estimates. Correctly distinguishing between confounders, mediators, and colliders allows researchers to avoid biases and uncover both direct and indirect causal pathways in complex systems.

An important application is within the exposome framework, where both the }external exposome}—representing life-course exposures such as environmental factors or lifestyle—and the internal exposome—reflecting an individual's current biological state—jointly contribute to disease development. Given the assumption of temporality, the external exposome precedes and influences the internal exposome, which in turn affects disease risk. Using SEM and DAGs, you will learn to estimate the direct and indirect effects of these exposures, addressing both the total impact and the pathways through which they operate.

This practical will deepen your understanding of causal inference while equipping you with the skills to apply these techniques in high-dimensional molecular epidemiological data, where temporal and causal relationships are key to understanding disease etiology. Importantly, because we cannot fully rule out residual unmeasured confounding, the estimated direct and indirect effects should not be interpreted as definitive causal estimates. Instead, they represent attributable risks contingent on the covariates included in our model.

This practical is divided into three sections:

1.Causal Effect Estimation - In this practical, we will begin with a simple simulated example to investigate the impact of conditioning on a confounder, a mediator, and a collider on effect estimation. We will also illustrate the concepts of the back-door and front-door criteria.

2.Sufficient Adjustment Set Determination - Next, we will utilize more complex directed acyclic graphs (DAGs) to determine the necessary adjustment set required for estimating total, direct, and indirect effects of a given variable.

3.High-Dimensional Mediation Analysis - Finally, we will identify minimally sufficient adjustment sets in a high-dimensional context using a reticular action model or a series of LASSO regressions, implemented through the sharp package.

Causal Effect Estimation

Confounder, Mediator, Collider

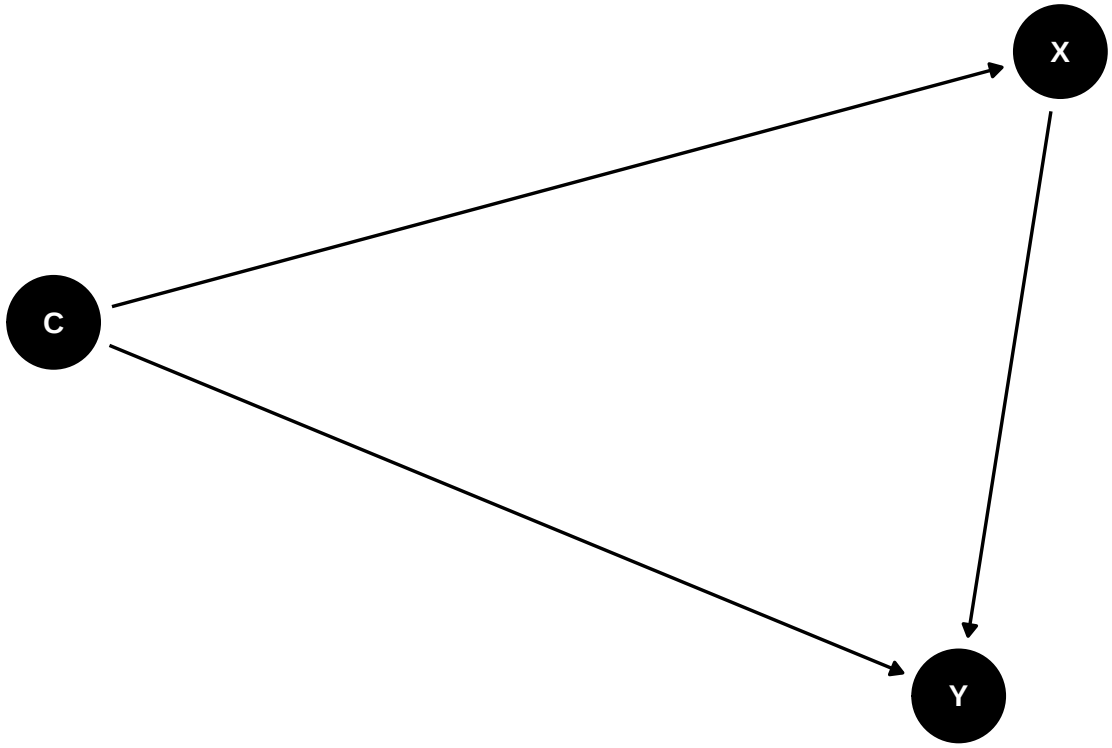
```
library(ggplot2)
library(dagitty)
library(ggdag)
```

```
##
## Attaching package: 'ggdag'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
# Confounder DAG
dag_confounder <- dagitty("dag { X -> Y; C -> X; C -> Y }")
ggdag(dag_confounder, text = TRUE) + theme_dag_blank() +
    ggtitle("Confounder DAG")
```

Confounder DAG



```r
set.seed(123)
n <- 1000
C <- rnorm(n)
X <- 0.5 * C + rnorm(n)
Y <- 0.5 * C + 0.5 * X + rnorm(n)

# Confounder Model
model_unadjusted <- lm(Y ~ X)
model_adjusted <- lm(Y ~ X + C)

summary(model_unadjusted)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0739 -0.6871 -0.0049  0.7569  3.2637
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02348    0.03344  -0.702    0.483
## X            0.72632    0.02875  25.265   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 998 degrees of freedom
## Multiple R-squared:  0.3901, Adjusted R-squared:  0.3895
## F-statistic: 638.3 on 1 and 998 DF,  p-value: < 2.2e-16
```

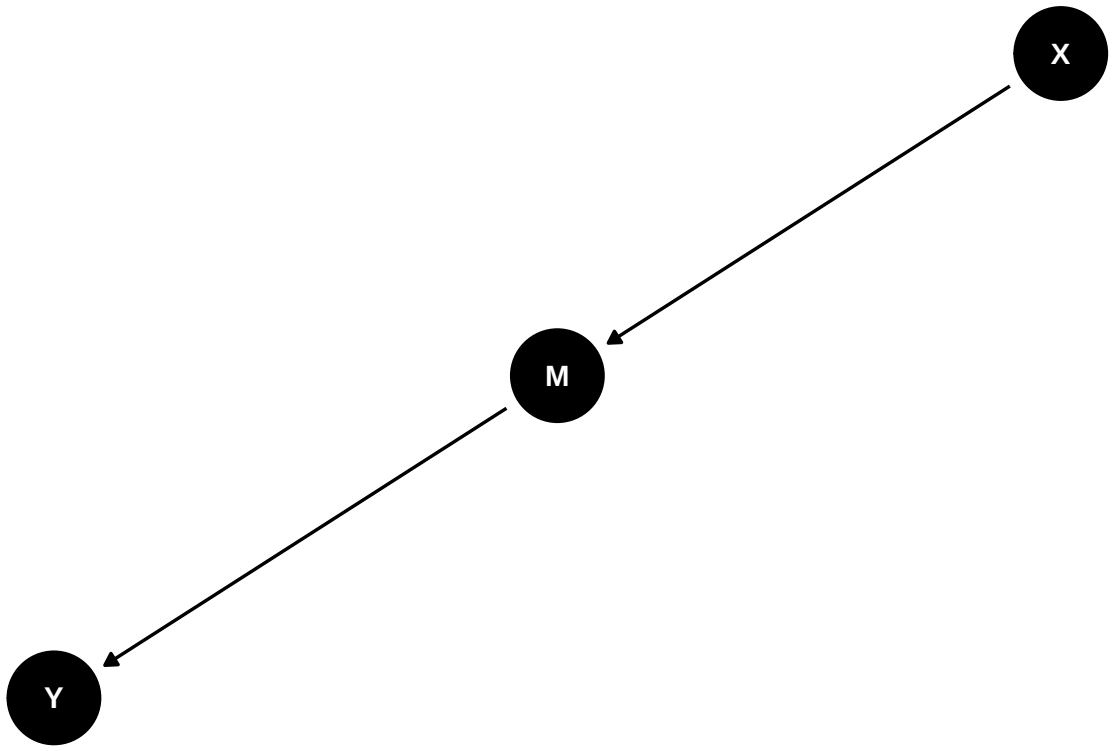
```r
summary(model_adjusted)
```

```
##
## Call:
## lm(formula = Y ~ X + C)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.8360 -0.6277 -0.0370  0.6538  3.3787
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02093    0.03098  -0.676    0.499
## X            0.52751    0.03079  17.135   <2e-16 ***
## C            0.46476    0.03609  12.876   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9788 on 997 degrees of freedom
## Multiple R-squared:  0.4771, Adjusted R-squared:  0.476
## F-statistic: 454.8 on 2 and 997 DF,  p-value: < 2.2e-16
```

```r
# Mediator DAG
dag_mediator <- dagitty("dag { X -> M -> Y }")
ggdag(dag_mediator, text = TRUE) + theme_dag_blank() +
    ggtitle("Mediator DAG")
```

Mediator DAG



```r
set.seed(123)
M <- 0.5 * X + rnorm(n)
Y <- 0.5 * M + rnorm(n)

# Mediator Model
model_unadjusted <- lm(Y ~ X)
model_adjusted <- lm(Y ~ X + M)

summary(model_unadjusted)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -6.54e-15 -1.36e-16 -9.00e-17 -6.00e-18  6.26e-14
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -7.477e-18  6.327e-17 -1.180e-01    0.906
## X            1.250e+00  5.439e-17  2.298e+16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.999e-15 on 998 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.281e+32 on 1 and 998 DF,  p-value: < 2.2e-16
```

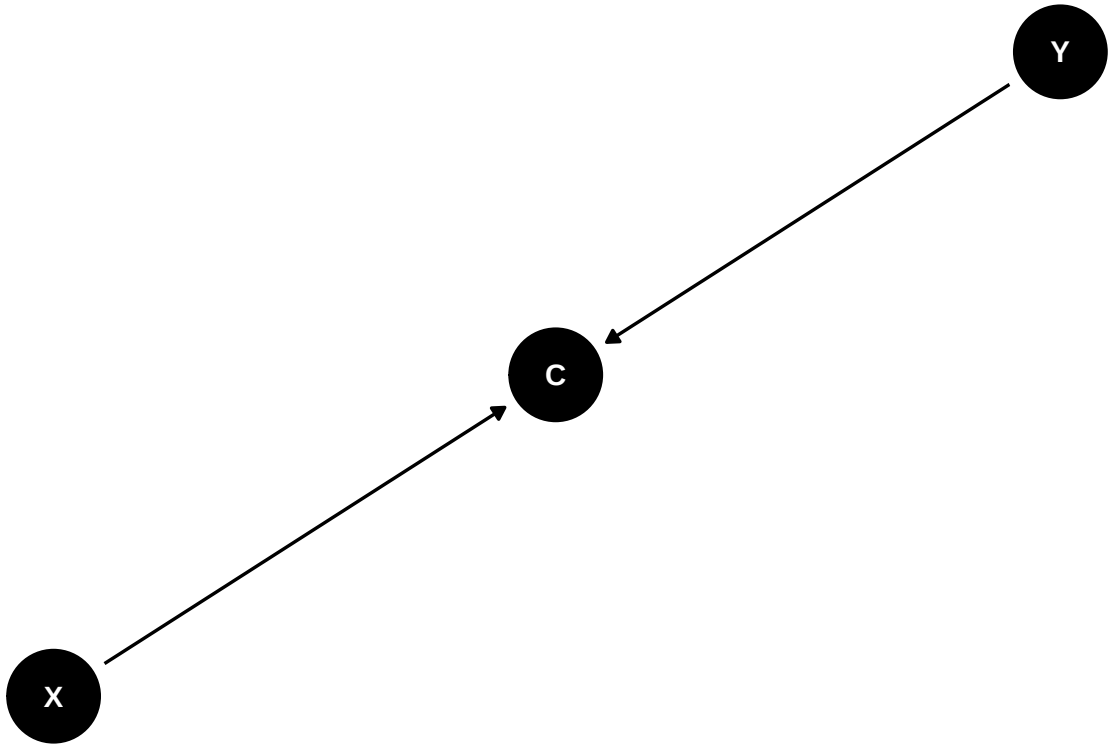
```r
summary(model_adjusted)
```

```
##
## Call:
## lm(formula = Y ~ X + M)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -6.54e-15 -1.41e-16 -9.00e-17 -1.00e-18  6.26e-14
##
```

```
## Coefficients:
##               Estimate Std. Error   t value Pr(>|t|)
## (Intercept) -7.477e-18  6.330e-17 -1.18e-01    0.906
## X            1.250e+00  8.742e-17  1.43e+16   <2e-16 ***
## M            2.663e-17  7.375e-17  3.61e-01    0.718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2e-15 on 997 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.638e+32 on 2 and 997 DF,  p-value: < 2.2e-16
```

```r
# Collider DAG
dag_collider <- dagitty("dag { X -> C; Y -> C }")
ggdag(dag_collider, text = TRUE) + theme_dag_blank() +
    ggtitle("Collider DAG")
```

Collider DAG




```r
set.seed(123)
C <- 0.5 * X + 0.5 * Y + rnorm(n)
```

```r
# Collider Model
model_unadjusted <- lm(Y ~ X)
model_adjusted <- lm(Y ~ X + C)

summary(model_unadjusted)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -6.54e-15 -1.36e-16 -9.00e-17 -6.00e-18  6.26e-14
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -7.477e-18  6.327e-17 -1.180e-01    0.906
## X            1.250e+00  5.439e-17  2.298e+16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.999e-15 on 998 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.281e+32 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
summary(model_adjusted)
```

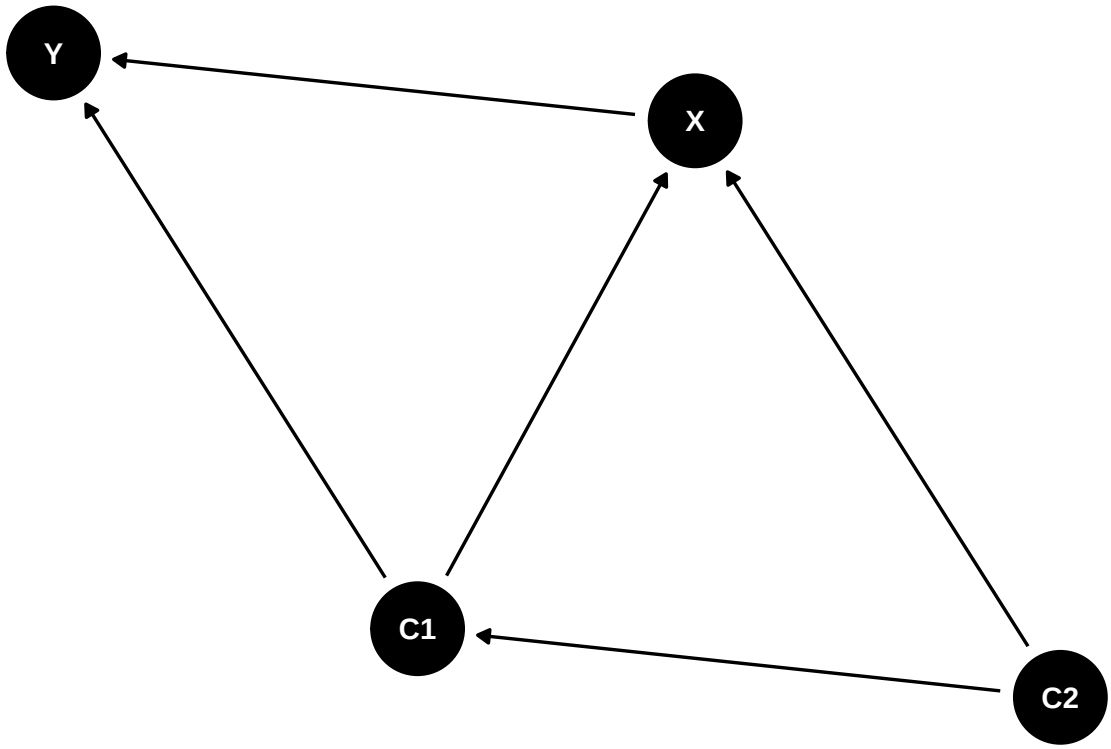
```
##
## Call:
## lm(formula = Y ~ X + C)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -6.54e-15 -1.41e-16 -9.00e-17 -1.00e-18  6.26e-14
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -7.477e-18  6.330e-17 -1.180e-01    0.906
## X            1.250e+00  1.268e-16  9.859e+15   <2e-16 ***
```

```
## C            2.663e-17  7.375e-17  3.610e-01    0.718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2e-15 on 997 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.638e+32 on 2 and 997 DF,  p-value: < 2.2e-16
```

Back-door criterion

```
# Correct Backdoor DAG
dag_backdoor_correct <- dagitty("dag {
  X -> Y
  C1 -> X
  C1 -> Y
  C2 -> C1
  C2 -> X
}")

# Plot the DAG
ggdag(dag_backdoor_correct) + theme_dag_blank() + ggtitle("Correct Backdoor Criterion DAG (C1 I
```

Correct Backdoor Criterion DAG (C1 Blocks the Path)




```r
set.seed(123)
n <- 1000

# Generate confounders C1 and C2
C2 <- rnorm(n)  # C2 only affects C1
C1 <- 0.6 * C2 + rnorm(n)  # C1 affected by C2
X <- 0.5 * C1 + 0.2 * C2 + rnorm(n)  # X affected by C1
Y <- 0.7 * X + 0.6 * C1 + rnorm(n)  # Y affected by X and C1, but not by C2

# Unadjusted Model (without controlling for
# confounders)
model_unadjusted <- lm(Y ~ X)
summary(model_unadjusted)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.2547 -0.7788  0.0089  0.7787  3.2485
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01861    0.03587   0.519    0.604
## X            1.08278    0.02921  37.063   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.134 on 998 degrees of freedom
## Multiple R-squared:  0.5792, Adjusted R-squared:  0.5788
## F-statistic:  1374 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
# Adjusted Model (with C1 as the only confounder)
model_adjusted_C1 <- lm(Y ~ X + C1)
summary(model_adjusted_C1)
```

```
##
## Call:
## lm(formula = Y ~ X + C1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.06524 -0.63409  0.00442  0.67809  2.90054
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.007724   0.031408  -0.246    0.806
## X            0.750727   0.031808  23.602   <2e-16 ***
## C1           0.563513   0.032142  17.532   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9921 on 997 degrees of freedom
## Multiple R-squared:  0.6784, Adjusted R-squared:  0.6777
## F-statistic:  1051 on 2 and 997 DF,  p-value: < 2.2e-16
```

```r
# Adjusted Model (with both C1 and C2)
model_adjusted_C1_C2 <- lm(Y ~ X + C1 + C2)
summary(model_adjusted_C1_C2)
```

```
##
## Call:
## lm(formula = Y ~ X + C1 + C2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.05617 -0.63344  0.00047  0.67526  2.89225
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.007759   0.031424  -0.247    0.805
## X            0.751481   0.032116  23.399   <2e-16 ***
## C1           0.566140   0.035519  15.939   <2e-16 ***
## C2          -0.006724   0.038616  -0.174    0.862
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9925 on 996 degrees of freedom
## Multiple R-squared:  0.6784, Adjusted R-squared:  0.6774
## F-statistic: 700.2 on 3 and 996 DF,  p-value: < 2.2e-16
```

Back-door Criterion By adjusting on C1 we block all backdoor paths between X and Y, creating a sufficient adjustment set.
The Backdoor Criterion ensures that there are no confounding biases when estimating the causal effect of X on Y. To apply the backdoor criterion, a set of variables (called the backdoor adjustment set) must be identified to block all backdoor paths — paths that connect X and Y through confounders, excluding the direct causal path from X to Y.
Backdoor Criterion Conditions:

1. No Direct Causal Path: The variable(s) you adjust for must not be descendants of X or Y.
2. Block All Backdoor Paths: A sufficient adjustment set must block all backdoor paths (i.e., paths from X to Y through confounders). A backdoor path is any path from X to Y that goes backward through a confounder.
3. Not a Collider: Do not adjust for a collider (a variable where two arrows collide). Adjusting for a collider can open a new path, creating collider bias.

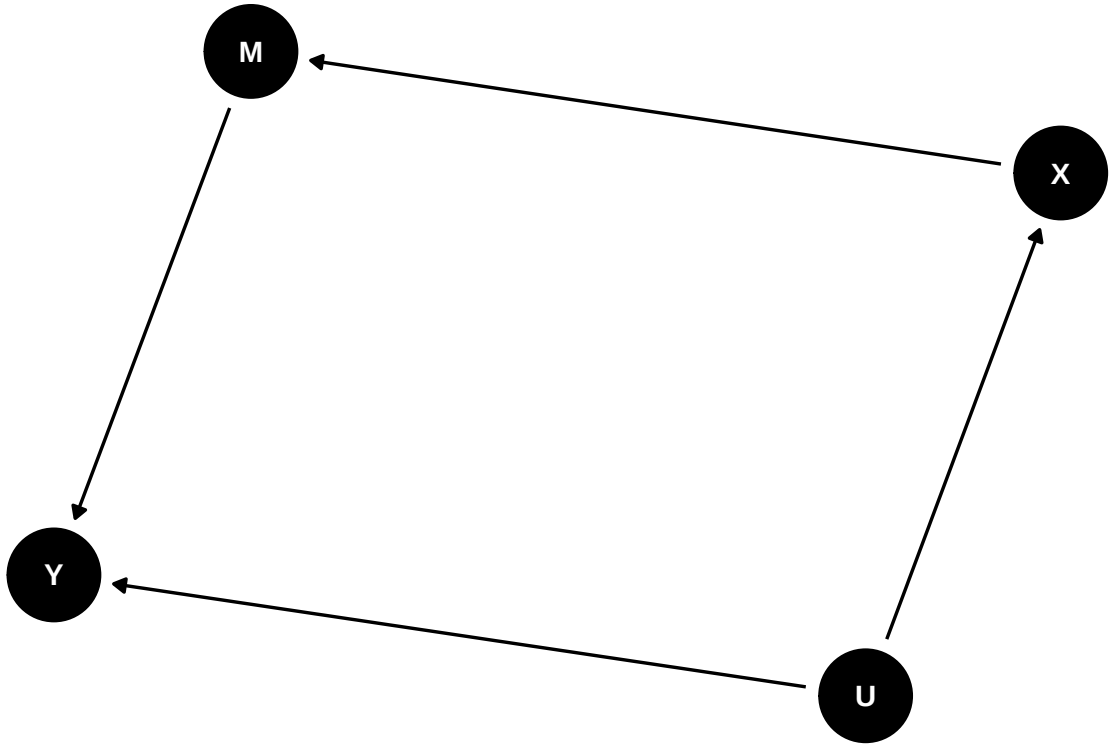
Approach:

- Identify the set of variables that can block all backdoor paths between X and Y to ensure an unbiased estimation of the causal effect.
- The backdoor criterion is often satisfied by adjusting for confounders — common causes of both X and Y.

Front-door criterion

```
# Frontdoor DAG
dag_frontdoor <- dagitty("dag {
  U -> X
  U -> Y
  X -> M
  M -> Y
}")

ggdag(dag_frontdoor) + theme_dag_blank() + ggtitle("Frontdoor Criterion DAG")
```

Frontdoor Criterion DAG




```
# Simulation
set.seed(123)
U <- rnorm(n)  # Unmeasured confounder
X <- 0.6 * U + rnorm(n)  # X influenced by U
M <- 0.5 * X + rnorm(n)  # Mediator M influenced by X
Y <- 0.7 * M + 0.6 * U + rnorm(n)  # Y influenced by both M and U

# Frontdoor Adjustment Step 1: Estimate the
```

```r
# effect of X on M
model_XM <- lm(M ~ X)
summary(model_XM)
```

```
##
## Call:
## lm(formula = M ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8334 -0.6387 -0.0287  0.6631  3.4342
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02064    0.03098  -0.666    0.505
## X            0.51010    0.02548  20.022   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9788 on 998 degrees of freedom
## Multiple R-squared:  0.2866, Adjusted R-squared:  0.2859
## F-statistic: 400.9 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
# Step 2: Estimate the effect of M on Y
model_MY <- lm(Y ~ M)
summary(model_MY)
```

```
##
## Call:
## lm(formula = Y ~ M)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8279 -0.7601 -0.0005  0.7799  3.5485
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0005338  0.0359851  -0.015    0.988
## M            0.8762483  0.0310846  28.189   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.138 on 998 degrees of freedom
## Multiple R-squared:  0.4433, Adjusted R-squared:  0.4427
## F-statistic: 794.6 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
# Step 3: Combine the effects from step 1 and
# step 2 to estimate the total effect
effect_X_on_Y_via_M <- coef(model_XM)["X"] * coef(model_MY)["M"]
effect_X_on_Y_via_M
```

```
##        X
## 0.446978
```

Front-door Criterion The Front-door Criterion allows for causal estimation when there are unmeasured confounders between X and Y, as long as there is a mediator that satisfies certain conditions. The mediator helps account for the indirect causal effect of X on Y, bypassing the unmeasured confounding.

Frontdoor Criterion Conditions:

1. Mediation: There exists a mediator M that lies on the causal pathway between X and Y (i.e., $X \to M \to Y$).
2. Unmeasured Confounding Between X and Y: There is an unmeasured confounder U affecting both X and Y, but U does not affect the mediator M. This ensures the effect of X on M remains unconfounded.
3. No Direct Path from X to Y (except through M): All effects of X on Y must go through M. There should be no direct causal effect from X to Y not captured by the mediator.
4. No Unmeasured Confounders Between M and Y: There must be no unmeasured confounders between the mediator M and the outcome Y.

Approach:

- The frontdoor criterion is useful when there is unmeasured confounding between X and Y, but a mediator M is available that explains the indirect effect of X on Y.
- By controlling for the mediator, you can estimate the causal effect even in the presence of unmeasured confounders.

Backdoor vs. Frontdoor Summary:

- Backdoor Criterion: Focuses on blocking backdoor paths (indirect connections through confounders) between X and Y to estimate the causal effect, by adjusting for confounders.
- Frontdoor Criterion: Allows estimation in cases with unmeasured confounding by leveraging a mediator that helps bypass the confounding through an indirect pathway from X to Y.

Sufficient Adjustment Set Determination

In groups of four, you will work together to apply the principles of the frontdoor and backdoor criteria to real-world research questions, build causal diagrams (DAGs), and identify sufficient adjustment sets for estimating total and direct effects.

Step 1: Develop a Research Question
As a group, brainstorm and agree on a real-world research question that involves causal relationships. Potentially use some of your own research.
Simple example research questions: - What is the causal effect of exercise on mental health? - How does education affect income, accounting for family background?

Step 2: Define Your Variable Set
Based on your research question, identify at least 5-6 key variables that are relevant to your causal inquiry. Ensure that you include potential confounders, mediators, colliders, and other relevant factors.
Example variables for the exercise and mental health question might include: - X = Exercise Frequency - Y = Mental Health - C_1 = Socioeconomic Status - C_2 = Age - M = Sleep Quality

Step 3: Construct a DAG
Using your assumptions, sketch a directed acyclic graph (DAG) that represents the relationships between your variables. Use causal arrows to depict which variables influence others. Include direct causal paths, potential confounders, mediators, and colliders.
Clearly label the nodes (variables) and arrows (causal relationships). You can use software tools such as DAGitty or pen and paper for this task.

Step 4: Apply the Backdoor and Frontdoor Criteria
Identify all possible backdoor paths between your exposure X and outcome Y. Based on these paths, determine a sufficient adjustment set to block confounding and estimate the total causal effect of X on Y.
Next, consider any possible mediators in your DAG and apply the frontdoor criterion. Determine the sufficient adjustment set for identifying the direct effect of X on Y, accounting for mediation.

Step 5: Challenge Each Other
Once each group has identified their adjustment sets, break into pairs within the group. One pair should present their adjustment sets for the total and direct effects, while the other pair challenges them by proposing alternative adjustment sets or questioning their assumptions about confounders, colliders, and mediators.
Rotate roles within the group so everyone has the opportunity to challenge the assumptions and explain their reasoning.

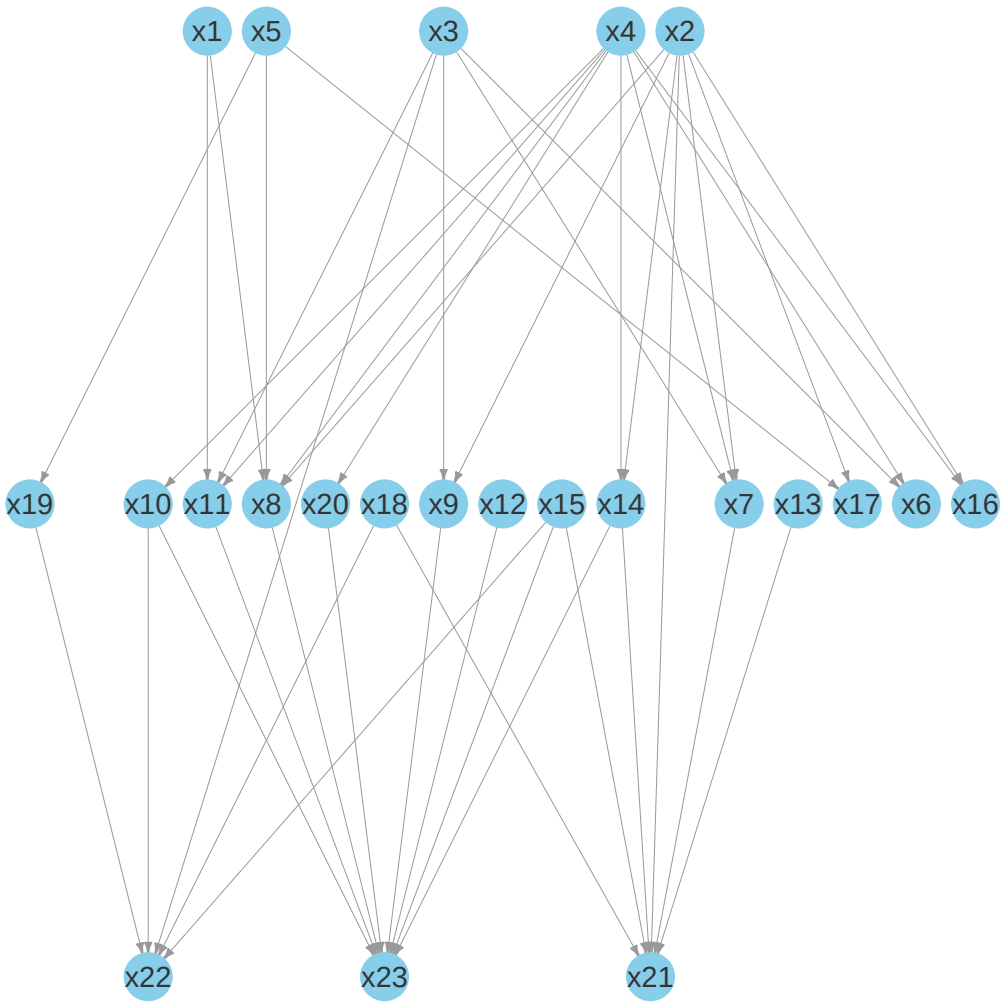
Discussion Questions: - Why did you choose certain variables as confounders or mediators? - Did adjusting for any variables open new bias-inducing paths (e.g., collider bias)? - Were there any surprises or disagreements in determining the adjustment sets? How did your group resolve them? - How would unmeasured confounders or mis-specified causal relationships affect your results?
Hint: You can check your result using the `adjustmentSets()` function in R, by specifying the `effect` argument you can derive adjustment sets for direct and total effects, respectively.

High-dimensional Mediation Analysis

On the basis of temporality, we hypothesize that exposure profiles cause specific molecular responses which in turn cause differences in disease risk. Hence, we have to investigate a) the direct effects of exposures and biomarkers on disease risk and b) the indirect effects of exposures on disease risk mediated via the measured biomarkers. Using a structural equation model (SEM) framework – a series of stability selection LASSOs, we can identify a) direct effects of exposures and biomarkers on disease risk, b) mediation effects of exposures on disease risk. We have to run two sets of models with different research questions; 1. For each disease, conditional on all variables, exposures and biomarkers, which sparse set is jointly and)stably) predictive of / causes a disease? (direct effects) 2. For each biomarker, which set of exposures is jointly (and stably) predictive of / causes the biological variable? (indirect effects)
As the fake package currently does not support simulating structural equation models with categorical variables, we will generate a study populagtion using continuous variables to illustrate the SEM framework. The principle remains the same.
Let us simulate a study population with n = 1000 individuals with 5 exposure variables, 15 biological features and 3 health outcomes. Each variable has an explained variance of 0.7 by its predictors. The `SimulateStructural()` function available in the fake package can be used to simulate data based on a Structural Equation Model. The argument `pk` denotes how many variables are in each layer, i.e. pk = c(5,15,3). The $nu\_between$ argument controls the probability that a variable is a true predictor.

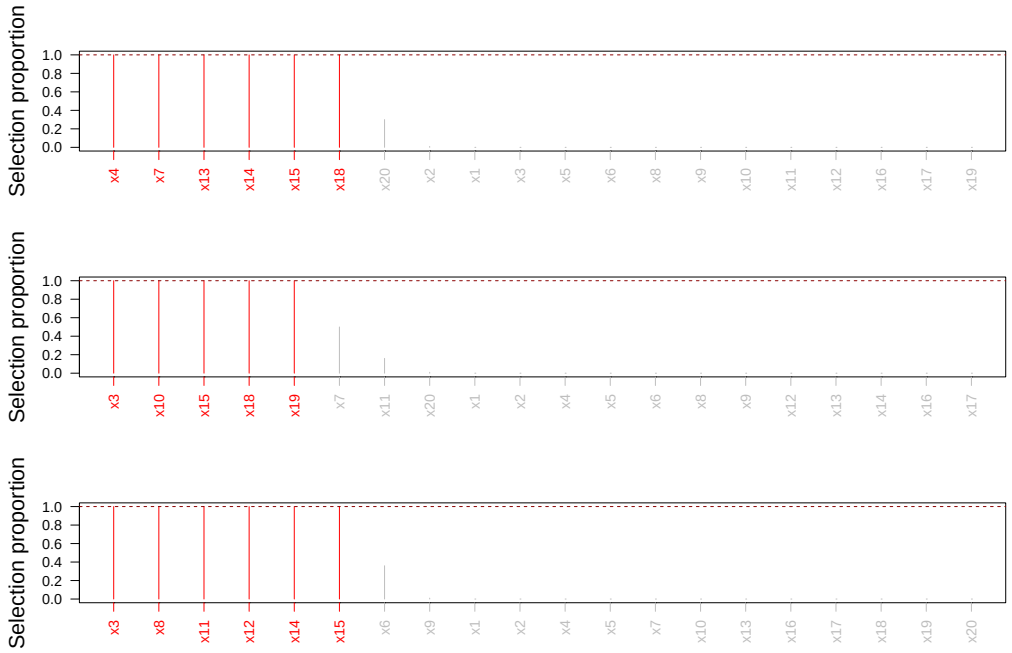
We eun the Structural Equation Model as a series of Stability Selected LASSO regressions described above (i.e. for each disease using Expotypes and Biomarkers as predictors, and for each Biomarker using Expotypes as predictors). List the variables selected in each model calibrated by maximizing the stability score.

```r
# Simulation of data with underlying structural
# causal model
library(sharp)
library(igraph)

set.seed(1)
pk = c(5, 15, 3)
simul_sem <- SimulateStructural(n = 1000, pk = pk,
    nu_between = 0.3, ev = 0.7)
plot(simul_sem, edge.arrow.size = 0.3)
```

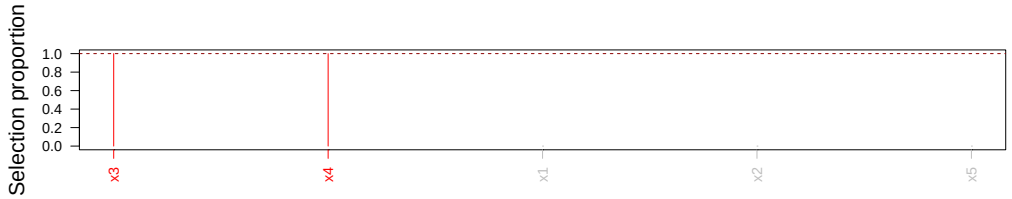


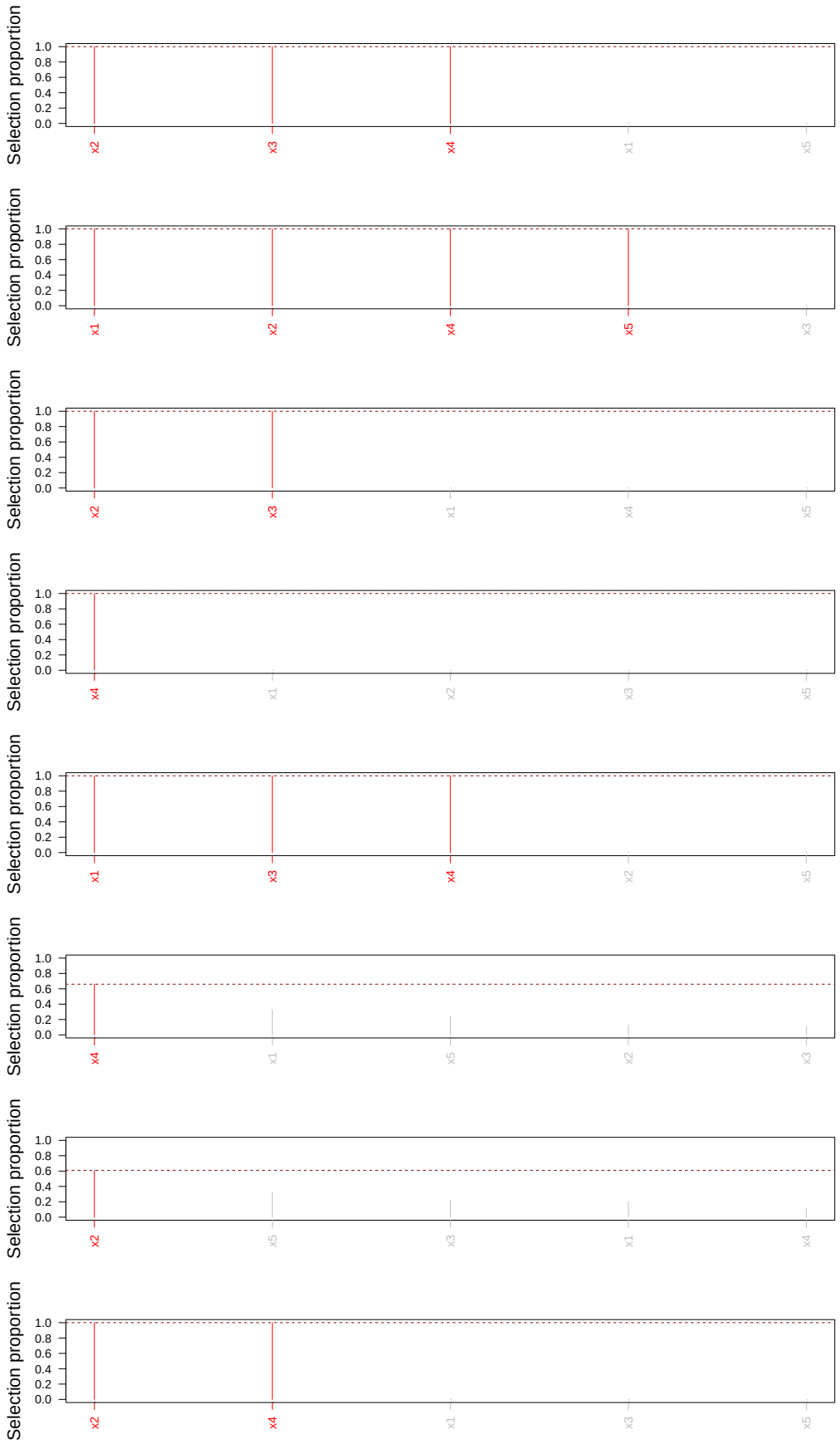

```r
## split the data into the Expotypes, Biological
## features and Health outcomes
simul_sem_expo <- simul_sem$data[, 1:5]
simul_sem_bio <- simul_sem$data[, 6:20]
simul_sem_health <- simul_sem$data[, 21:23]
```

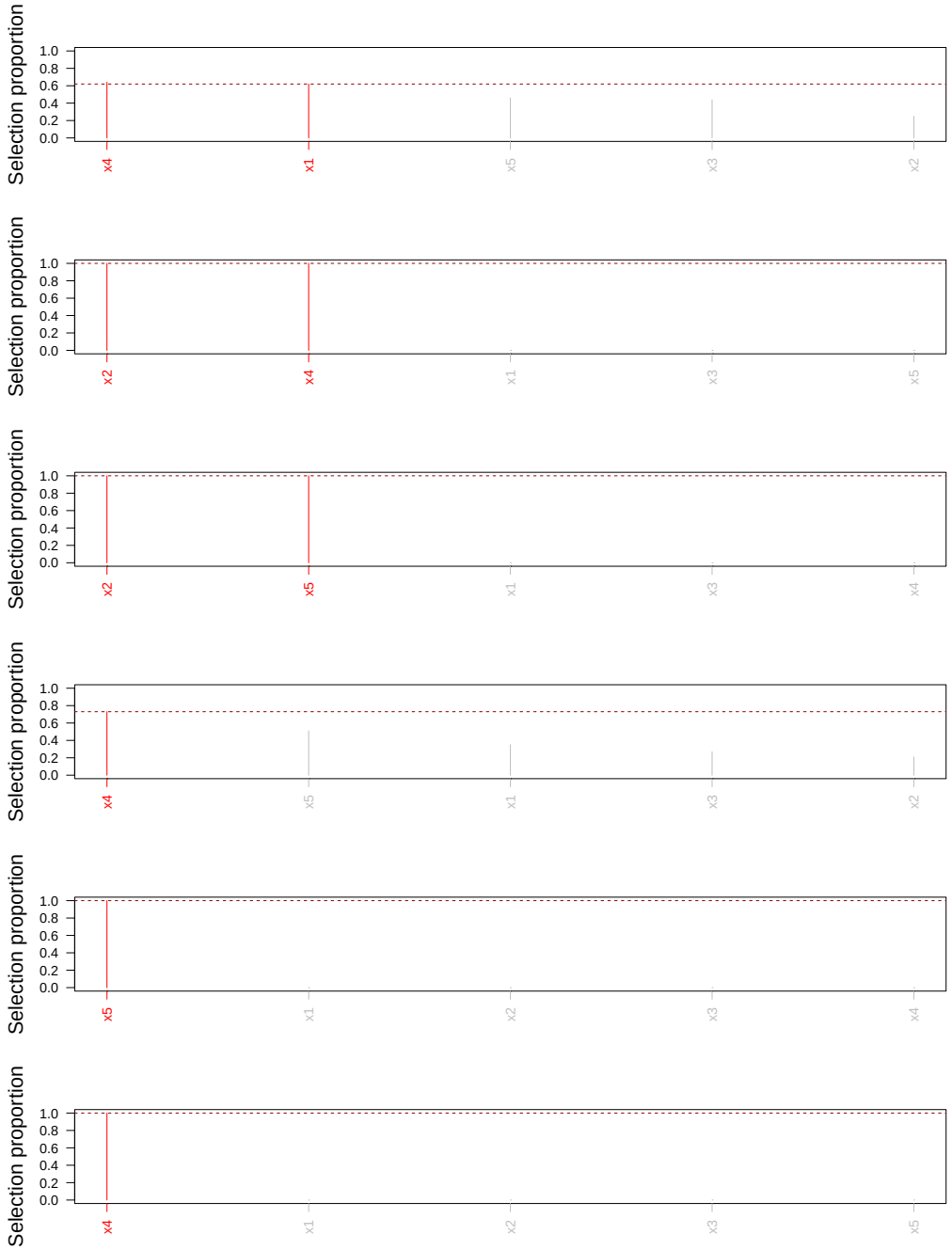
```r
# Model direct effects on diseases conditional on
# all other variables
stab_sem_direct <- apply(simul_sem_health, 2, FUN = function(y) {
    VariableSelection(xdata = cbind(simul_sem_expo,
        simul_sem_bio), ydata = y, n_cat = 3, pi_list = seq(0.5,
        1, by = 0.01))
})

lapply(stab_sem_direct, plot)
```



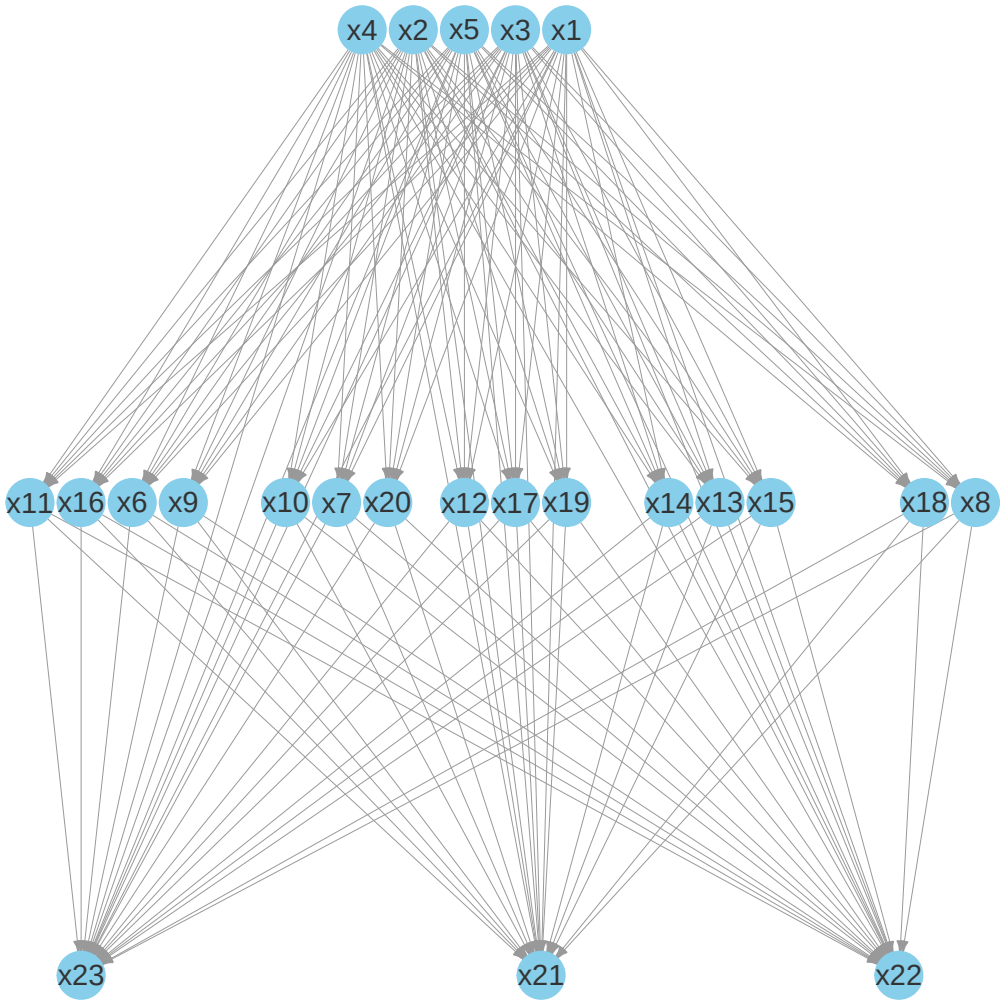

```r
# Model indirect effects of Expotypes on diseases
# by modelling direct effects of Expotypes on
# Biomarkers
stab_sem_indirect <- apply(simul_sem_bio, 2, FUN = function(y) {
    VariableSelection(xdata = simul_sem_expo, ydata = y,
        n_cat = 3, pi_list = seq(0.5, 1, by = 0.01))
})

lapply(stab_sem_indirect, plot)
```



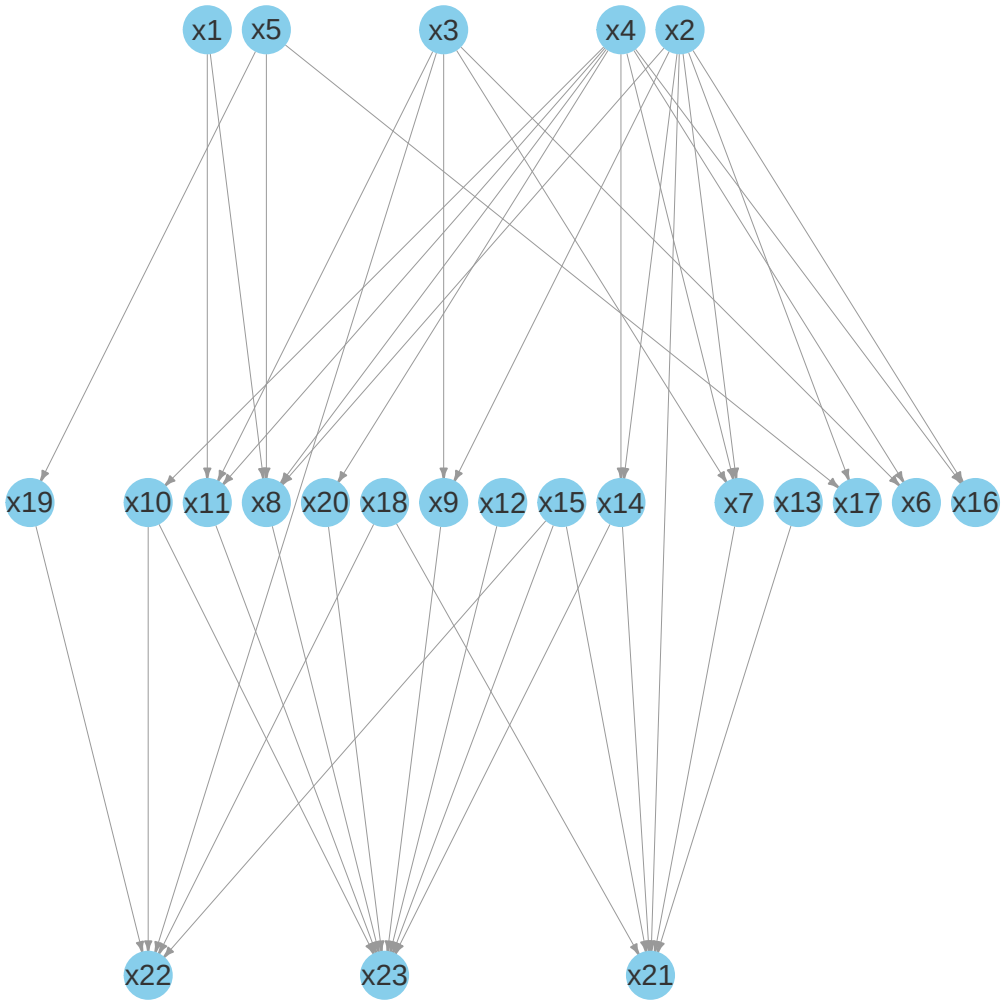

Selection proportion
1.0
0.8
0.6
0.4
0.2
0.0
x2 x3 x4 x1 x5
Selection proportion
1.0
0.8
0.6
0.4
0.2
0.0
x1 x2 x4 x5 x3
Selection proportion
1.0
0.8
0.6
0.4
0.2
0.0
x2 x3 x1 x4 x5
Selection proportion
1.0
0.8
0.6
0.4
0.2
0.0
x4 x1 x2 x3 x5
Selection proportion
1.0
0.8
0.6
0.4
0.2
0.0
x1 x3 x4 x2 x5
Selection proportion
1.0
0.8
0.6
0.4
0.2
0.0
x4 x1 x5 x2 x3
Selection proportion
1.0
0.8
0.6
0.4
0.2
0.0
x2 x5 x3 x1 x4
Selection proportion
1.0
0.8
0.6
0.4
0.2
0.0
x2 x4 x1 x3 x5



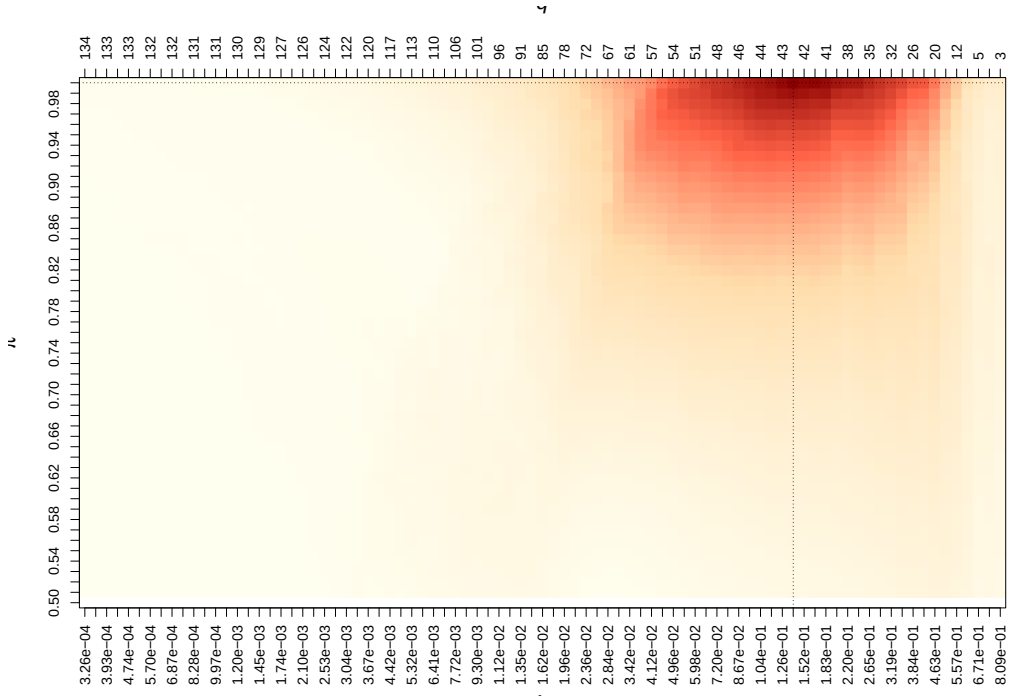

```
# Alternatively this edge-selection procedure can
# be run via the StructuralModel() function
dag <- LayeredDAG(pk)

plot(dag, edge.arrow.size = 0.3)
```



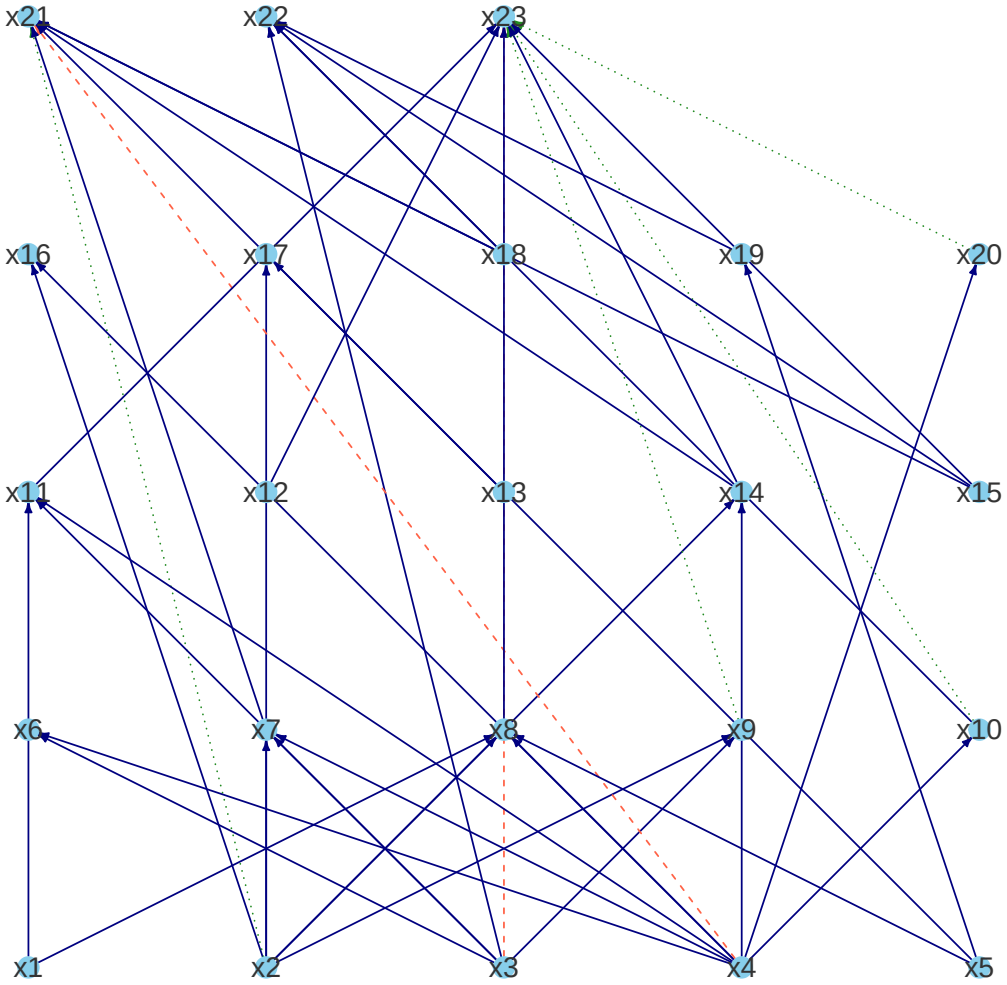

```
plot(simul_sem, edge.arrow.size = 0.3)
```




```
stab_sem <- StructuralModel(x = simul_sem$data, adjacency = dag,
    n_cat = 3, pi_list = seq(0.5, 1, by = 0.01))
CalibrationPlot(stab_sem)
```




```
plot(SelectionPerformanceGraph(Adjacency(stab_sem),
    simul_sem$theta), layout = layout.grid, edge.arrow.size = 0.3)
```


x21
x22
x23
x16
x17
x18
x19
x20
x11
x12
x13
x14
x15
x6
x7
x8
x9
x10
x1
x2
x3
x4
x5