

# Directed acyclic graphs: what are they and what are they useful for?

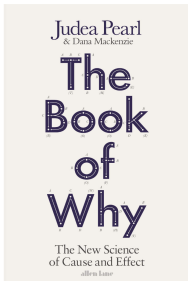
Tom Palmer, `tom.palmer@lancaster.ac.uk`

Lecturer in Medical Statistics, Department of Mathematics and Statistics, Lancaster  
University

Keele University, Tuesday 4<sup>th</sup> December 2018

- Introduction
- Introduction to directed acyclic graphs (DAGs)
- $d$ -separation rules
- Statistical independence
- Backdoor paths and confounding
- Examples
- Disadvantages of DAGs
- Discussion

- A lot of the theory developed in the late 1980s and 1990s (Pearl (1995))
- Hit mainstream only relatively recently (Munafò et al. (2018), Hernan (2017))
- Recent publicity, Pearl's Book of Why published this year



- Lots of interest in epidemiology, however . . . *DAG anxiety*



**george davey smith**

@mendel\_random

Follow



Scared of the DAG police, but using figures?  
Add: "This figure is a schematic representation and should not be interpreted as a formal DAG"

3:39 AM - 15 May 2017

6 Retweets 12 Likes



3



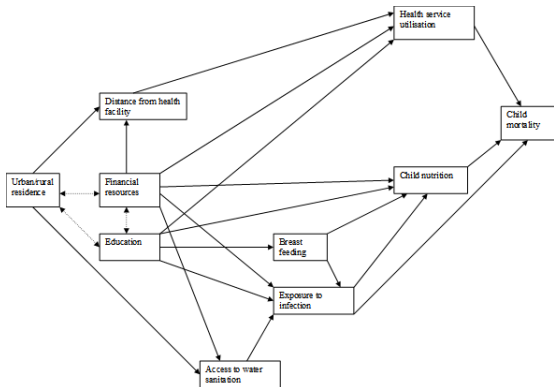
6



12

- The (old) rules of epidemiological modelling:
  - Adjust for common causes
  - Do not adjust for common effects
  - Do not adjust for variables on the causal pathway
- Easy to apply to simple situations with a few variables
- But how do we apply these when the model is (realistically) complex?

- What should we adjust/not adjust for to estimate the effect of Health service utilisation ON Child mortality?



## Causal

- A DAG is said to be causal for an effect if all common causes of the exposure and outcome are on the DAG

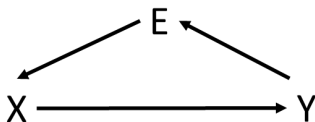
## D: Directed

- DAGs depict structural relationships – causal effects without modelling assumptions
- Unlike SEM path diagrams they do not show residuals
- DAGs for different models are the same, e.g.
  - linear regression of  $Y$  on  $X$ , logistic regression of  $Y$  on  $X$



## A: Acyclic

- Following the direction of arrows from  $X$  we should not be able to get back to  $X$
- This DAG is not allowed



- Intuition: the rules of conditional independence



## G: Graph

- Nodes represent variables
- Say we have this regression,

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$$

- Arrows represent effects (arrow from  $X$  to  $Y$  represents  $\beta_1$ )



## Path

- Any consecutive sequence of arrows (edges) regardless of the direction of the arrow

# *d*-separation rules I

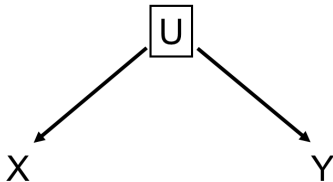
- *d*: directional
- Defined by Pearl (1995)
- Rule 1: if there are no variables being conditioned on, a path is blocked if and only if 2 arrows collide at some point on the path



- Because of the collider  $U$  we say  $X$  and  $Y$  are not *d*-connected
- *d*-connected: unblocked path between 2 variables (i.e. path with no collider)

# d-separation rules II

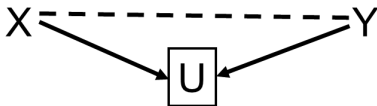
- Rule 2: Any path that contains a non-collider/common cause/confounder that has been conditioned on is blocked



- Conditioning/adjusted for/included in a model denoted by square box around variable

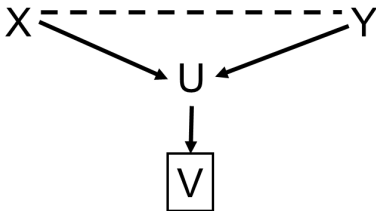
# *d*-separation rules III

- Rule 3: A collider that has been conditioned on does not block a path



# *d*-separation rules IV

- Rule 4: A collider that has a descendant that has been conditioned on does not block a path



# $d$ -separation rules V

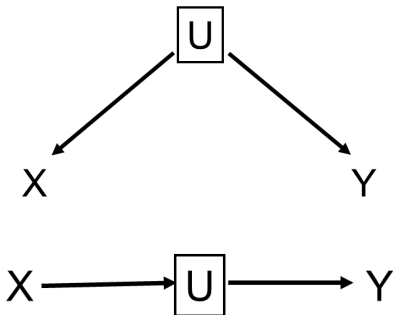
- If a pair of variables are  $d$ -separated they are statistically independent (conditional on any variables required to block backdoor paths between them)
- $X$  and  $Y$  independent

$X$

$Y$

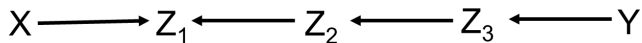
# *d*-separation rules VI

- $X$  and  $Y$  independent given  $U$ ; written as  $X \perp\!\!\!\perp Y|U$



# *d*-separation rules VII

- Complex pathways now have a (hopefully) clearer interpretation

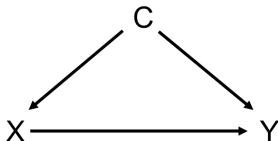


- Are  $X$  and  $Y$  *d*-separated if we condition on:
  - $Z_1$ ?
  - $Z_2$ ?



# Backdoor paths and confounding

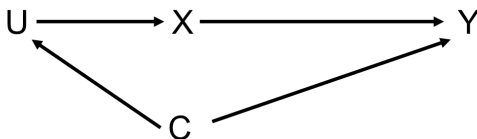
- A backdoor path starts by travelling the wrong way along an arrow



- X-C-Y is a backdoor path

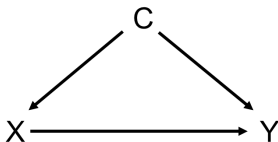
# Backdoor paths and confounding II

- We can travel the wrong direction along an arrow more than once



- X-U-C-Y is a backdoor path

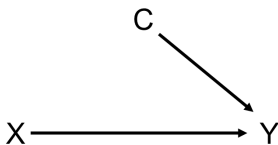
# Backdoor paths and confounding III



- Here  $C$  fulfills conventional definition of a confounder, because it is:
  - associated with  $X$  (arrow  $C$ - $X$ )
  - associated with  $Y$  conditional on  $X$  (arrow  $C$ - $Y$ )
  - is not on the causal pathway between  $X$  and  $Y$
- Structural definition of confounding: the existence of an open backdoor path between  $X$  and  $Y$ .

# Backdoor paths and confounding IV

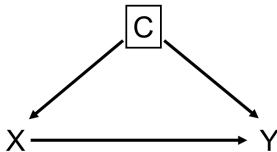
- In trials randomization removes C-X (X: randomized treatment) arrow



# Backdoor paths and confounding

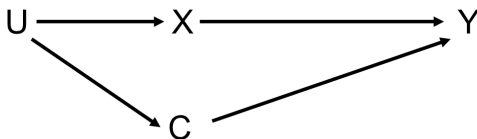
## V

- If we condition on  $C$  (e.g. include it as an additive covariate in model) then the path is blocked



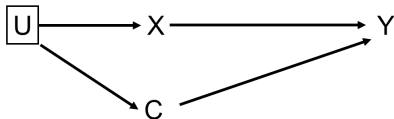
# Backdoor paths and confounding VI

- In this DAG, to estimate the effect of  $X$  on  $Y$  what should we do?

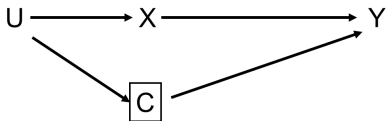


# Backdoor paths and confounding VII

- We can adjust for  $U$

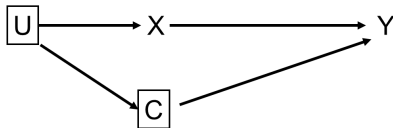


- We can adjust for  $C$



# Backdoor paths and confounding VIII

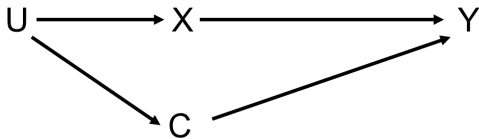
- We can adjust for both  $U$  and  $C$





# Backdoor paths and confounding IX

- Defining a variable as a confounder is relative to which effect we are estimating
- To estimate the effect of  $U$  on  $Y$  what should we do?



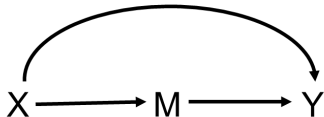
# Backdoor paths and confounding X

- Say we propose the model:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 m_i + \varepsilon_i, \quad \varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$$

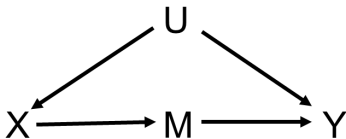


- If the estimate  $\hat{\beta}_1$  is found not to be null, we could have

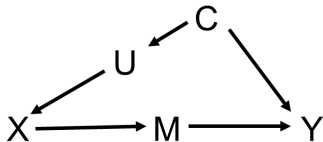


# Backdoor paths and confounding XI

- or

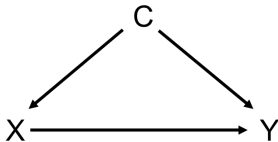


- or an even more complex confounding structure (or others)



# Confounding example I

- Let's investigate what happens when we simulate some data
- Assuming linear models



- True model is:  $y_i = \beta_0 + \beta_1 x_i + \beta_2 c_i + \varepsilon_i$ ,  $\varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$

# Confounding example II

- Of course the unadjusted model is biased

```
set.seed(123456)
n <- 150
c <- rnorm(n)
x <- c + rnorm(n)
y <- c + x + rnorm(n)
lm(y ~ x) %>% summary() %>% coef()
```

```
##              Estimate Std. Error   t value    Pr(>|t|)
## (Intercept) -0.04430578 0.10116577 -0.4379522 6.620594e-01
## x            1.40874066 0.06933222 20.3187003 1.164942e-44
```

# Confounding example III

- The true model is unbiased

```
lm(y ~ x + c) %>% summary() %>% coef()
```

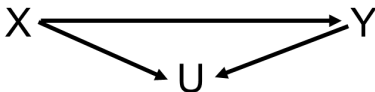
```
##              Estimate Std. Error    t value    Pr(>|t|)
## (Intercept) 0.001936471 0.08563298  0.02261361 9.819892e-01
## x           0.896588477 0.08808434 10.17875007 9.487072e-19
## c           1.008965959 0.12965362  7.78201172 1.164183e-12
```

```
lm(y ~ x + c) %>% confint.default()
```

```
##              2.5 %    97.5 %
## (Intercept) -0.1659011 0.169774
## x           0.7239463 1.069231
## c           0.7548495 1.263082
```

# Colliding example I

- If the DAG is



- Adjusting for  $U$  induces bias

```
set.seed(123456)
n <- 150
x <- rnorm(n)
y <- x + rnorm(n)
u <- x + y + rnorm(n)
lm(y ~ x + u) %>% summary() %>% coef()
```

# Colliding example II

```
##           Estimate Std. Error    t value    Pr(>|t|)
## (Intercept) 0.0428632 0.06130921  0.6991316 4.855745e-01
## x           0.1798452 0.10948445  1.6426553 1.025917e-01
## u           0.4611080 0.04530104 10.1787501 9.487072e-19
lm(y ~ x + u) %>% confint.default()
```

```
##           2.5 %    97.5 %
## (Intercept) -0.07730064 0.1630270
## x           -0.03474037 0.3944308
## u           0.37231957 0.5498964
```



# Colliding example III

- Fitting the correct model

```
lm(y ~ x) %>% summary() %>% coef()
```

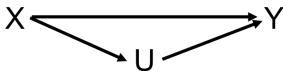
```
##           Estimate Std. Error    t value    Pr(>|t|)
## (Intercept) 0.07459585 0.07967631  0.9362362 3.506768e-01
## x           1.09975103 0.08041755 13.6755105 4.837012e-28
```

```
lm(y ~ x) %>% confint.default()
```

```
##           2.5 %    97.5 %
## (Intercept) -0.08156685 0.2307585
## x           0.94213554 1.2573665
```

# Mediation example I

- If the DAG is



- To estimate the direct effect of  $X$  on  $Y$

```
set.seed(123456)
n <- 150
x <- rnorm(n)
u <- x + rnorm(n)
y <- x + u + rnorm(n)
coef(summary(modelm <- lm(y ~ x + u)))
```

# Mediation example II

---

##		Estimate	Std. Error	t value	Pr(> t )
##	(Intercept)	0.001936471	0.08563298	0.02261361	9.819892e-01
##	x	1.008965959	0.12965362	7.78201172	1.164183e-12
##	u	0.896588477	0.08808434	10.17875007	9.487072e-19

# Mediation example III

- To estimate the total effect we fit

```
lm(y ~ x) %>% summary() %>% coef()
```

```
##           Estimate Std. Error   t value   Pr(>|t|)
## (Intercept) 0.06881825  0.1111027  0.6194113 5.365978e-01
## x           1.99499006  0.1121363 17.7907659 1.379348e-38
```

# Mediation example IV

- To estimate the indirect effect, first fit

```
coef(summary(mod1 <- lm(u ~ x)))
```

```
##              Estimate Std. Error    t value    Pr(>|t|)
## (Intercept) 0.07459585 0.07967631  0.9362362 3.506768e-01
## x           1.09975103 0.08041755 13.6755105 4.837012e-28
```

- Multiply the X-U and U-Y path coefficients

```
(indefeff <- coef(mod1)[2] * coef(modelm)[3])
```

```
##           x
## 0.9860241
```

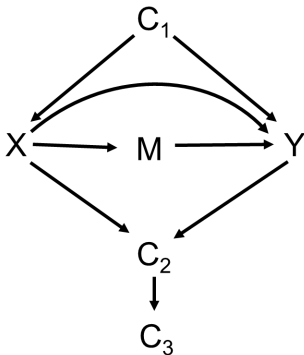
- Compare sum of direct and indirect effects to previous total effect

```
(totoeff <- coef(modelm)[2] + indefeff)
```

```
##           x
## 1.99499
```

# More complex example I

- If the DAG is



# More complex example II

```
set.seed(123456)
n <- 150
c1 <- rnorm(n)
x <- c1 + rnorm(n)
m <- x + rnorm(n)
y <- c1 + x + m + rnorm(n)
c2 <- x + y + rnorm(n)
c3 <- c2 + rnorm(n)
```

# More complex example III

- For the direct effect of  $X$  on  $Y$  of course the simple model is biased

```
coef(summary(modelx <- lm(y ~ x)))
```

##	Estimate	Std. Error	t value	Pr(> t )
## (Intercept)	0.07280811	0.12618137	0.5770116	5.648083e-01
## x	2.37476121	0.08647624	27.4614312	5.805841e-60



# More complex example IV

- Adjusting for  $C_1$  and  $M$  recovers the direct effect as expected

```
lm(y ~ x + m + c1) %>% summary() %>% coef()
```

##	Estimate	Std. Error	t value	Pr(> t )
## (Intercept)	0.1115822	0.07946580	1.404154	1.623967e-01
## x	1.0173919	0.10672719	9.532640	4.795425e-17
## m	1.0107777	0.07653848	13.206138	1.073170e-26
## c1	0.8796623	0.12031788	7.311151	1.609465e-11

# More complex example V

- Adjusting for either  $C_2$  or  $C_3$  or both induces bias

```
lm(y ~ x + m + c1 + c2) %>% summary() %>% coef()
```

##	Estimate	Std. Error	t value	Pr(> t )
## (Intercept)	0.06511859	0.05925209	1.0990092	2.735855e-01
## x	-0.03128823	0.12467456	-0.2509592	8.022009e-01
## m	0.47786867	0.07501430	6.3703681	2.349755e-09
## c1	0.34870404	0.10186449	3.4232148	8.046834e-04
## c2	0.52432123	0.04807025	10.9073949	1.337696e-20

# More complex example VI

- Adjusting for  $C_3$

```
lm(y ~ x + m + c1 + c3) %>% summary() %>% coef()
```

##	Estimate	Std. Error	t value	Pr(> t )
## (Intercept)	0.1080981	0.06351287	1.701987	9.090087e-02
## x	0.2389449	0.12053160	1.982425	4.932154e-02
## m	0.6772346	0.07122761	9.508036	5.841841e-17
## c1	0.5322631	0.10339923	5.147650	8.442526e-07
## c3	0.3557591	0.03891790	9.141270	5.073325e-16

# More complex example VII

- Adjusting for both  $C_2$  and  $C_3$

```
coef(summary(fullmodel <- lm(y ~ x + m + c1 + c2 + c3)))
```

##	Estimate	Std. Error	t value	Pr(> t )
## (Intercept)	0.07529289	0.05863517	1.2840910	2.011723e-01
## x	-0.04845372	0.12324295	-0.3931561	6.947856e-01
## m	0.49102547	0.07424300	6.6137609	6.835023e-10
## c1	0.35649550	0.10055884	3.5451432	5.297073e-04
## c2	0.39564302	0.07470576	5.2960174	4.336211e-07
## c3	0.12546383	0.05627968	2.2292920	2.734301e-02

# More complex example VIII

- Model selection algorithms do not perform well
- They assume the covariates are either independent predictors or confounders
- Backwards selection

```
library(MASS)
```

```
##  
## Attaching package: 'MASS'  
  
## The following object is masked from 'package:dplyr':  
##  
##      select  
stepAIC(fullmodel, direction = "backward", scope = list(lower = modelx),  
         trace = 0)
```

# More complex example IX

```
##  
## Call:  
## lm(formula = y ~ x + m + c1 + c2 + c3)  
##  
## Coefficients:  
## (Intercept)          x          m          c1          c2  
##    0.07529    -0.04845    0.49103    0.35650    0.39564  
##           c3  
##    0.12546
```

# More complex example X

- Forwards selection

```
fwd <- step(modelx, scope = list(lower = modelx, upper = fullmodel),  
          direction = "forward", trace = 0)  
coef(fwd)
```

```
## (Intercept)          x          c2          m          c1          c3  
## 0.07529289 -0.04845372 0.39564302 0.49102547 0.35649550 0.12546383
```

# More complex example XI

- Both directions selection

```
bth <- step(modelx, scope = list(lower = modelx, upper = fullmodel),  
           direction = "both", trace = 0)  
bth$coefficients
```

```
## (Intercept)          x          c2          m          c1          c3  
## 0.07529289 -0.04845372 0.39564302 0.49102547 0.35649550 0.12546383
```

- Pearl's Simpsons Machine example

<http://www.dagitty.net/learn/simpson/index.html>

- Stepwise inclusion of covariates changes the sign of the effect at every step



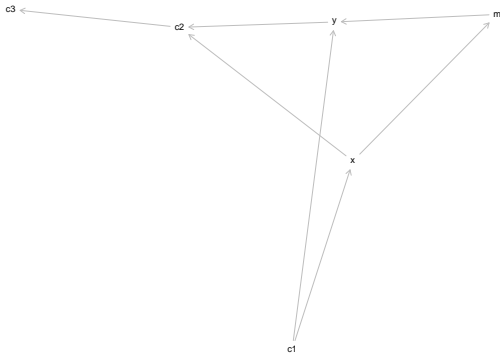
# More complex example XII

- We can use DAGitty <http://www.dagitty.net/> to help us (Textor et al. (2016))

```
library(dagitty)
complexg <- dagitty("dag {
    x -> m -> y
    x <- c1 -> y
    x -> c2
    y -> c2
    c2 -> c3
}")
```

# More complex example XIII

```
plot(graphLayout(complexg))
```



# More complex example XIV

- List testable implications

```
impliedConditionalIndependencies(complexg) %>% print()
```

```
## c1 _||_ c2 | x, y  
## c1 _||_ c3 | c2  
## c1 _||_ c3 | x, y  
## c1 _||_ m | x  
## c2 _||_ m | x, y  
## c3 _||_ m | x, y  
## c3 _||_ m | c2  
## c3 _||_ x | c2  
## c3 _||_ y | c2  
## x _||_ y | c1, m
```

# More complex example XV

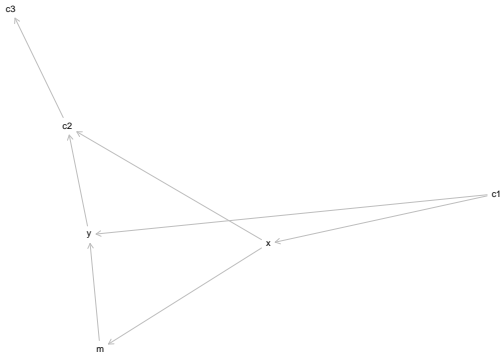
- Equivalence class
- Two DAGs are Markov equivalent if they represent the same conditional independence relations

```
eqdags <- equivalentDAGs(complexg)  
length(eqdags)
```

```
## [1] 3
```

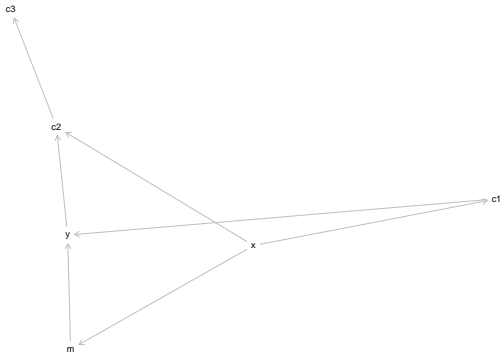
# More complex example XVI

```
plot(graphLayout(eqdags[[1]]))
```



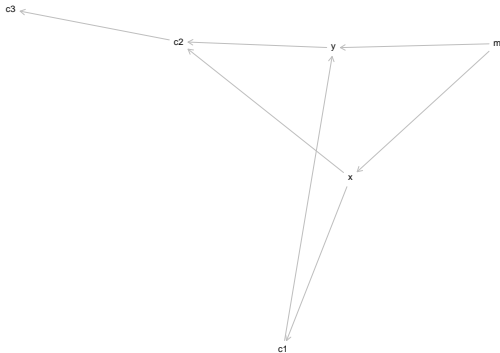
# More complex example XVII

```
plot(graphLayout(eqdags[[2]]))
```



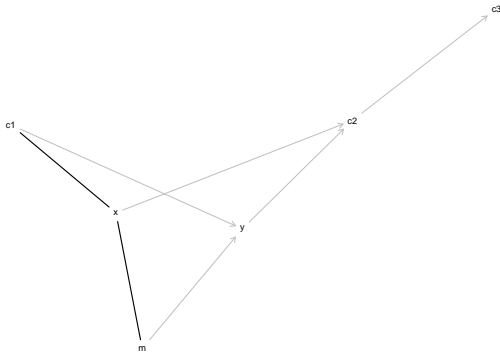
# More complex example XVIII

```
plot(graphLayout(eqdags[[3]]))
```



# More complex example XIX

```
eqclass <- equivalenceClass(complexg)  
plot(graphLayout(eqclass))
```





# More complex example XX

- Number of edges that can be reversed without changing the equivalence class

```
sum(edges(equivalenceClass(complexg))$e == "--")
```

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

# More complex example XXI

- List adjustment sets for effect of interest

```
adjustmentSets(complexg, "x", "y", effect = "direct" ) %>% print()
```

```
## { c1, m }
```

- List adjustment sets for total effect

```
adjustmentSets(complexg, "x", "y") %>% print()
```

```
## { c1 }
```

# More complex example XXII

```
for(n in names(complexg)){  
  for( m in setdiff(descendants(complexg, n ), n)){  
    a <- adjustmentSets(complexg, n, m)  
    if(length(a) > 0 ){  
      cat("The total effect of ",n," on ",m,  
          " is identifiable controlling for:\n", sep = "")  
      print(a, prefix=" * ")  
    }  
  }  
}
```

```
## The total effect of c1 on y is identifiable controlling for:  
## * {}  
## The total effect of c1 on c2 is identifiable controlling for:  
## * {}  
## The total effect of c1 on c3 is identifiable controlling for:  
## * {}  
## The total effect of c1 on x is identifiable controlling for:  
## * {}  
## The total effect of c1 on m is identifiable controlling for:  
## * {}  
## The total effect of c2 on c3 is identifiable controlling for:  
## * {}  
## The total effect of m on y is identifiable controlling for:  
## * { c1 }  
## * { x }
```

# More complex example XXIII

```
## The total effect of m on c2 is identifiable controlling for:  
## * { x }  
## The total effect of m on c3 is identifiable controlling for:  
## * { x }  
## The total effect of x on m is identifiable controlling for:  
## * {}  
## The total effect of x on y is identifiable controlling for:  
## * { c1 }  
## The total effect of x on c2 is identifiable controlling for:  
## * { c1 }  
## The total effect of x on c3 is identifiable controlling for:  
## * { c1 }  
## The total effect of y on c2 is identifiable controlling for:  
## * { x }  
## * { c1, m }  
## The total effect of y on c3 is identifiable controlling for:  
## * { x }  
## * { c1, m }
```

# More complex example XXIV

---

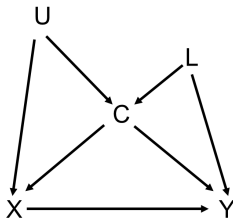
## DAGitty's missing feature

- Pass it correlations between a set of variables (correlations) and generate all possible DAGs
  - Apparently this is being developed

# Disadvantages of DAGs I

- They do not tell us the functional form of the model
  - What outcome model do we fit, e.g. linear regression/logistic regression etc.?
  - What parametric form should our variables have, e.g.  $X$ ,  $X^2$
- If there is confounding bias – we don't know if important
- If there is colliding bias – we don't know if important

# Disadvantages of DAGs II



- We may even need to trade-off confounding and colliding biases:
  - If  $U$  and  $L$  both unmeasured
  - Not adjusting for  $C$  – confounding bias
  - Adjusting for  $C$  – colliding bias
  - We don't know which is worse without doing simulations for our example

# Disadvantages of DAGs III

- If we decompose our effect with a mediator we don't know relative sizes of direct and indirect effects
- Difficult to represent interactions on a DAG
- Most realistically complex modelling situations will probably generate multiple plausible DAGs



- Modelling guidelines informed by causal DAGs
  - Adjust for a set of variables sufficient to block all backdoor pathways between the two variables of interest
  - Do not adjust for colliders or variables caused by colliders
  - If a variable is on the causal pathway adjusting for it will decompose the effect of interest
- Thanks for your attention
- Any questions

Hèrnan, Miguel. 2017. “Causal Diagrams: Draw Your Assumptions Before Your Conclusions.” <https://www.edx.org/course/causal-diagrams-draw-assumptions-harvardx-ph559x>.

Munafò, Marcus R, Kate Tilling, Amy E Taylor, David M Evans, and George Davey Smith. 2018. “Collider Scope: When Selection Bias Can Substantially Influence Observed Associations.” *International Journal of Epidemiology* 47 (1): 226–35.  
<https://doi.org/10.1093/ije/dyx206>.

Pearl, Judea. 1995. “Causal Diagrams for Empirical Research.” *Biometrika* 82 (4). [Oxford University Press, Biometrika Trust]: 669–88. <http://www.jstor.org/stable/2337329>.

# References II

---

Textor, Johannes, Benito van der Zander, Mark S Gilthorpe, Maciej Liškiewicz, and George TH Ellison. 2016. “Robust Causal Inference Using Directed Acyclic Graphs: The R Package ‘Dagitty’.” *International Journal of Epidemiology* 45 (6): 1887–94.  
<https://doi.org/10.1093/ije/dyw341>.