Article:
Textor, J, van der Zander, B, Gilthorpe, MS orcid.org/0000-0001-8783-7695 et al. (2 more authors) (2017) Robust causal inference using directed acyclic graphs: the R package 'dagitty'. International Journal of Epidemiology, 45 (6). pp. 1887-1894. ISSN 0300-5771

https://doi.org/10.1093/ije/dyw341

Robust causal inference using Directed Acyclic Graphs: the R package 'dagitty'

Johannes Textor[1], Benito van der Zander[2], Mark S. Gilthorpe[3,4], Maciej Liśkiewicz[2], and George T.H. Ellison[3,4]

[1]Department of Tumour Immunology, Radboud University Medical Center, P.O. Box 9101, 6500 HB Nijmegen, The Netherlands
[2]Institute for Theoretical Computer Science, University of Luebeck, Ratzeburger Allee 160, 23562 Luebeck, Germany
[3]Division of Epidemiology & Biostatistics, Leeds Institute of Cardiovascular and Metabolic Medicine, School of Medicine, University of Leeds, Worsley Building, Clarendon Way, Leeds, LS2 9LU, UK
[4]Leeds Institute for Data Analytics, University of Leeds, LS2 9NL

Abstract

Directed Acyclic Graphs (DAGs), which offer systematic representations of causal relationships, have become an established framework for the analysis of causal inference in epidemiology; often being used to determine covariate adjustment sets for minimizing confounding bias. DAGitty is a popular web application for drawing and analysing DAGs. Here we introduce the R package 'dagitty', which provides access to all of the capabilities of the DAGitty web application within the R platform for statistical computing, and also offers several new functions. We describe how the R package 'dagitty' can be used to: evaluate whether a DAG is consistent with the dataset it is intended to represent; enumerate 'statistically equivalent' but causally different DAGs; and identify exposure-outcome adjustment sets that are valid for causally different but statistically equivalent DAGs. This functionality enables epidemiologists to detect causal misspecifications in DAGs and make robust inferences that remain valid for a range of different DAGs.

Availability

The R package 'dagitty' is available through the comprehensive R archive network (CRAN) at https://cran.r-project.org/web/packages/dagitty/. The source code is available on github at https://github.com/jtextor/dagitty. The web application 'DAGitty' is free software, licensed under the GNU general public license (GPL) version 2 and is available at http://dagitty.net/.

Introduction

Greenland et al.'s seminal article (1), in which they describe a range of systematic representations of causal relationships that facilitate the specification of statistical analyses, is widely credited with introducing directed acyclic graphs (DAGs) to the field of epidemiology. Since then, DAGs have grown in popularity and have been included in popular epidemiology textbooks (2). One of the most attractive features of DAGs is that they provide principled procedures for identifying suitable sets of covariates for removing structural confounding bias through adjustment (1,3,4), using graphical criteria such as the so-called 'back-door' criterion (5) and its extensions (6–8). While these criteria are intuitive to apply in DAGs containing few variables, they become cumbersome to use in those with larger numbers of variables – a situation that is not uncommon in many epidemiological studies. The challenge of working with larger DAGs containing more than a handful of variables is what motivated the development of the web application DAGitty (9). This application contains graphical

tools for drawing DAGs and automated algorithms capable of rapidly specifying all minimal sufficient adjustment sets. To-date the DAGitty application has been cited by more than 100 empirical studies to support causal inference analyses of observational data, including a recent article published in this journal which used DAGitty to explore the possible role of serum bilirubin levels in the development of hypertension (10).

The decision to develop DAGitty as a web application was based primarily on accessibility considerations and a desire to facilitate its use across a range of different computing platforms. However, most quantitative epidemiological research involves analyses performed using dedicated statistical software (such as Stata, SAS or R). Situating DAGitty as a web application therefore requires epidemiologists to use separate software for analysing models generated using DAGs – as was the case in a recent study reported in the *European Psychiatry Journal* (11), which used the DAGitty web application and the SAS software package. For these analyses it would be more efficient to have DAGitty's functionality embedded within the statistical software used. Indeed, the lack of integration between the DAGitty web application and standard statistical software may have discouraged researchers from using DAGs.

DAGs are often viewed as purely qualitative causal path diagrams, which only make claims about the presence and absence of causal effects, not about the strength of such effects. Yet, most DAGs actually impose implicit quantitative restrictions on the probability distributions of the datasets with which they are compatible. These restrictions emerge as a consequence of the "d-separation" property (a more detailed explanation of d-separation is available as Supplementary data at *IJE* online). As these restrictions are open to statistical evaluation, it is possible to assess formally whether the restrictions imposed by any given DAG hold in the dataset(s) the DAG is intended to represent. In other words, provided the graphical criteria used to identify covariate adjustment sets have been correctly applied, the appropriateness of these will depend on whether the DAG itself has been correctly specified – thus formal statistical evaluation of DAG-dataset consistency is a potentially powerful tool for identifying many of the errors in the way a DAG has been specified.

There is therefore a compelling case for developing a single platform that combines graphical tools for drawing DAGs with statistical tools for evaluating the restrictions these DAGs impose on the datasets they are intended to represent. To address this, the R package 'dagitty' was created. This package not only provides direct access to all the features of the original DAGitty web application from *within* R, but also contains several novel features that are not (yet) available in the web application. Rather than describing all of these new features here, the present report will instead focus on those features that are relevant to covariate adjustment sets – this currently being one of the most attractive features of DAGs within epidemiology. To this end, in addition to explaining how the package helps evaluate DAG-dataset consistency, the present report will demonstrate how the package can also be used to identify covariate adjustment sets that are robust to a number of common misspecifications of causal paths within DAGs (such as causal paths that, as specified, are actually operating in the wrong direction).

Implementation

The R package 'dagitty' uses the same library of software routines that underlies the original DAGitty web application. This library is written in JavaScript and it is integrated into R by means of the package 'V8' by Jeroen Ooms (12), together with a set of wrapper routines written in R. This setup was chosen because the DAGitty library has been under continual development since 2010 and, as such, has achieved

a higher level of quality than could be expected from re-implementation in an alternative programming language. This setup is also intended to ensure that the web application and the R package can remain synchronized without the need to port features back and forth. As such, these advantages outweigh any potential limitations of non-native implementation. Importantly, from a user perspective, this architectural choice has no negative consequences for utility or function, since communication between R and the JavaScript library is handled internally by the package and is invisible to the user. The JavaScript library handles all tasks related to graph analysis, such as identification of adjustment sets. The data analysis functions, such as the statistical testing of DAG-implied restrictions, are implemented purely in R.

Usage

To demonstrate some of the more important functions of the R package 'dagitty' it is worth considering an example that reflects the way the DAGitty web application is typically used in epidemiology: a researcher drawing a DAG to determine which covariate adjustment set(s) are required to remove structural confounding bias (4,13); and therefore which covariates should be measured/included (see Figure 1A). Once these variables have been identified, and data on these collected, the R package 'dagitty' can then be used to evaluate whether the DAG (as specified *a priori*) is consistent with the dataset.

The R package 'dagitty' represents graphs by means of simple textual syntax, which strongly resembles the syntax of the software "graphviz" (14). This syntax has several features that allow graphs to be generated comprehensively so that most simple DAGs, with five or fewer variables, can be written in a single line of code. For example, consider the relatively simple DAG X→M→Y (a DAG known as the "full mediation model" because the causal effect of variable X on variable Y only occurs via the "mediation variable" M). In the textual syntax used by the R package 'dagitty', this DAG would simply be written as "dag { X -> M -> Y }". Furthermore, instead of typing the syntax of a DAG into the R package 'dagitty' by hand, it is possible to build the DAG using the DAGitty web application and then copy and paste (or download directly) the resulting syntax directly into an R script.

Evaluating DAG-dataset consistency

As alluded to earlier, the evaluation of DAG-dataset consistency draws on the statistically testable restrictions that emerge as a consequence of the so-called "d-separation" property. Testable restrictions can be found in any DAG that contains pairs of variables with no *direct* causal path between them (though not, unfortunately, in those DAGs where *all* the variables are pairwise linked by arrows – a phenomenon that may be common in some contexts, such as purely social or biosocial pathways). The d-separation property imposes restrictions in the form of conditional or unconditional independencies that *must* hold in any dataset that is generated by the causal process described by the DAG. For instance, the "full mediation model" (X→M→Y) implies through d-separation that X and Y *must* be conditionally independent given M (commonly written as $X \perp Y \mid M$). By testing such implications statistically, it is possible to evaluate whether the DAG, as specified, is consistent with the dataset it is intended to represent. If at least one implied independence does not hold in the dataset, then this means that the causal processes encoded by the DAG *cannot* have generated these data. If, instead, we test several implied independencies and none are refuted by the data, then this will lend credibility to the hypotheses encoded in the DAG, even though (as for any statistical test) these tests alone cannot *prove* that the DAG is correct. Moreover, it is important to recognise

that statistical tests of d-separation implications are *not* tests of 'null hypotheses', but are direct tests of the restrictions imposed by the DAG. Nonetheless, to avoid any potential confusion, this approach might be better described as *DAG-dataset consistency evaluation*.

A relatively simple strategy for testing any given conditional independence statement (such as $X \perp Y \mid Z$) is to regress both X and Y on Z, and then test for a non-zero correlation between the residuals (15). Where linear regression is used, this approach is equivalent to a test of zero partial correlation. For jointly normally distributed variables, conditional independence implies zero partial correlation. However, for non-normal data, the partial correlation can be non-zero even when the variables examined are conditionally independent. In such instances, non-parametric regression techniques should be used to compute the residuals. The R package 'dagitty' currently supports both linear regression and local polynomial regression to compute residuals, offering both parametric and semi-parametric tests of conditional independence, respectively.

To illustrate how this approach might be applied, Figure 1 gives an example based on sports medicine exploring the possible causal relationship between "performance of warm-up exercises" (WUE) as the exposure on "injury" (I) as the outcome (3). In this example, the DAG (as specified by the research team) is missing a direct arrow from "team motivation" (TM) to "performance of warm-up exercises" (WUE; Figure 1A). To evaluate DAG-dataset consistency the DAG model code (from the DAGitty web application) is copied and pasted, or downloaded directly, into an R script in which the dataset is also uploaded (the only stipulation being that the names of variables used when specifying the DAG are the same as those in the dataset; Figure 1B). The function 'localTests' is then used to apply the d-separation criterion and enumerate the DAG's implied conditional independencies, followed by a formal test of zero (partial) correlation for each of the identified independencies. The dedicated function 'plotLocalTestResults' visualizes the results of these tests using a plot of the empirical partial correlation coefficients and their confidence intervals (Figure 1C). As a rule, the farther from zero the empirical correlation, the less 'consistent' the corresponding implication with the dataset collected.

While this approach is relatively straightforward for DAGs containing only a few variables, larger DAGs can have many testable implications (the DAG in Figure 1A, for example, has 64). This means that the problem of multiple testing can become an issue. To mitigate this problem, the p-values obtained should be corrected for multiple testing. Various methods for p-value correction are available in R including the Holm-Bonferroni method (a more powerful version of the Bonferroni method, in which the k^{th} smallest of m p-values is multiplied by [m+1-k]). Importantly, the Holm-Bonferroni method does not assume independence of the hypotheses being tested. Example code to illustrate how the p-value correction for multiple testing is performed can be found in the Supplementary data at *IJE* online, and the results of applying this correction to the example considered earlier, is summarised in Figure 1C. This shows the three empirical correlations, each relating to separate testable implications, for which the corrected p-value is smaller than an arbitrary cut-off value of 0.05.

In those instances where implications of the type "X and Y are independent given M" are found to be inconsistent with the dataset collected, several potential reasons might need to be considered, including: (i) misspecification of relationships amongst the measured ('observed') variables included in the DAG (such as when the direction of an arrow on one, or more, of the causal paths has been misspecified *a priori*); (ii) omission of an unmeasured ('latent') variable within the DAG that is a common cause

of two or more other variables (such that, although there is no direct causal link between the two, they are nonetheless correlated); and/or (iii) measurement error in one or more of the variables included (whether by chance/at random, or where a latent variable causes measurement error in the two variables, as in (ii) above). All of these potential reasons require careful consideration in the light of the best available knowledge of established functional/causal relationships amongst the variables therein.

In the example DAG summarised in Figure 1, all three of the testable implications that were found to be inconsistent with the dataset relate to the same two variables: "team motivation" (TM); and "warm-up exercises" (WUE) – as a result of the missing direct arrow from the former to the latter. In this instance, the research team should therefore reconsider whether their decision to omit this direct arrow was correct, and where there is no compelling substantive theory to support the omission of this causal path, the DAG should be revised with the arrow between the two included. In this instance, the inclusion of the arrow between TM and WUE would change the adjustment sets applicable to the exposure-outcome of interest (i.e. the relationship between WUE and "injury" (I); Figure 1D), and reduce the number of minimal adjustment sets from four to three. Note, however, that all three minimal adjustment sets for the relationship between WUE and I in the revised DAG are also valid (though not necessarily *minimal*) for the original (i.e. unrevised) DAG.

Valid adjustment sets for statistically equivalent DAGs

It is worth restating that, while consistency between any given DAG and the dataset it is intended to represent might bolster confidence in the hypotheses encoded therein, this in itself does not amount to 'proof' that the DAG is correct. Indeed, an important limitation of this approach is that different DAGs *can* have exactly the same testable implications. For example, the DAG for the "full mediation model" described earlier ($X \rightarrow M \rightarrow Y$) has exactly the same testable implication ($X \perp Y \mid M$) as both: the DAG in which $X \leftarrow M \leftarrow Y$ (i.e. the symmetrically opposite scenario, where Y causes X entirely mediated through M); and the DAG in which $X \leftarrow M \rightarrow Y$ (i.e. where there is no causal path in either direction between X or Y, but instead both are caused by M). All three of these DAGs imply that X and Y are independent given M, even though their functional/causal interpretations are very different. For this reason, the R package 'dagitty' includes additional functions that help to identify and evaluate different DAGs that have exactly the same testable implications.

The function 'equivalentDAGs' generates a list of all possible DAGs that are statistically equivalent to the DAG originally specified. For example, as shown in Figure 1E, there are five other DAGs that are equivalent to the DAG shown in Figure 1A (a so-called "equivalence class" of DAGs). Equivalence classes are purely based on the testable implications, and therefore, they are not dependent on the exposure(s) and outcome(s) for which the DAG was originally intended. However, adjustment sets *do* depend upon the exposure and outcome specified. For example, only five of the six DAGs in the equivalence class shown in Figure 1E share the same minimal sufficient adjustment sets as the original DAG for the relationship between WUE and I.

In those instances where the *same* minimal sufficient adjustment sets apply to an exposure-outcome relationship in *all* of the DAGs in an equivalence class, this greatly strengthens confidence that these sets are valid (especially if the DAG has also undergone DAG-dataset consistency evaluation, as described earlier). The R package 'dagitty' can identify such cases using a recently published generalized version of the back-door criterion (8,16). To demonstrate this feature within a real world example,

this has been applied to a DAG derived from the recently published bilirubin/hypertension study (10). This fairly complex DAG (see Figure 2A) has a total of 41 arrows, and its equivalence class contains 40 DAGs. The 'equivalenceClass' function provides a useful graphical summary of such large equivalence classes, in which any arrows that have the *same* direction in *all* of the different DAGs are displayed normally (i.e. "←" or "→"), while arrows with *different* directions in *different* DAGs are displayed without arrowheads (i.e. "–"; Figure 2B). There are 30 arrows in the bilirubin DAG whose direction is the same in the entire equivalence class; therefore, an error in the direction of any of these arrows would lead to a change in the testable implications, and is therefore potentially detectable using DAG-dataset consistency evaluation (as described earlier). In contrast, errors in the direction of any of the remaining 11 arrows may or may not lead to a change in the testable implications, depending on whether the resulting DAG is still in the equivalence class. However, by combining the functions 'equivalenceClass' and 'adjustmentSets' it is possible to determine that, in this DAG, the same minimal sufficient adjustment set would remain valid for the entire equivalence class for the exposure-outcome relationship under investigation (i.e. between bilirubin and hypertension; see Figure 2C). Therefore, for this DAG, the issue of statistical equivalence would not be a concern for the validity of the adjustment set determined. The same conclusion would be drawn for 51 out of 136 possible exposure-outcome relationships that can be investigated for this DAG (Figure 2D).

In this way, the R package 'dagitty' makes it possible to identify robust minimal sufficient adjustment sets by combining the evaluation of DAG-dataset consistency with the identification of valid adjustment sets for statistically equivalent DAGs.

Conclusion

The present report introduced two key functions of the R package 'dagitty': the first, evaluating the consistency of DAGs with the datasets they are intended to represent; and the second, deriving covariate adjustment sets that are valid for whole groups of different (but statistically equivalent) DAGs. Other features of the R package 'dagitty', which are related to instrumental variables and the testing of linear structural equation models, are described in more detail within the reference manual accompanying the package. For those studying causal inference, there is also an R vignette (available at http://dagitty.net/primer/), which shows how to solve many of the exercises in the recent textbook on causality by Pearl et al. (17).

While these new tools help strengthen confidence in the use of DAGs in a range of specific circumstances, it is important to point out that one should avoid the temptation to use evaluations of DAG-dataset inconsistency to generate *purely* data-driven, post-hoc modifications to DAGs. This runs the risk of 'over-fitting' and biased inference in which: the (modified) DAG is no longer specified *a priori* on the basis of established functional/causal relationships between variables; and the consistency of these DAG-specified relationships with the dataset the DAG was intended to represent can no longer be evaluated (since the DAG has been modified to 'fit' the dataset rather than specified on the basis of established functional/causal relationships).

Regarding the evaluation of DAG-dataset consistency, it is worth conceding that an important limitation of the current approach lies in the use of partial correlations to evaluate conditional independencies. This approach requires a normal distribution of the variables involved. Non-normality can be mitigated to some extend by using non-parametric regression to generate residuals (as described earlier; 15). However, non-parametric testing of conditional independence remains challenging and, motivated by

the growing importance of DAGs to the design of quantitative analyses, this has now become a research topic in its own right. A future aspiration is therefore to incorporate the results of advances in this area (18,19) within the R package 'dagitty' to provide a fuller range of analytical solutions.

Nevertheless, we believe that the functionality of the R package 'dagitty' will help address one key concern that has been raised about DAG methodology, namely that this approach "assume[s] that all … DAGs have been properly specified" (20). While it is true that the validity of DAG-based analyses, as for any statistical analyses, depends upon the validity of any underlying assumptions, many have welcomed the use of DAGs precisely because they help to reveal a number of these assumptions in a transparent and explicit fashion, so that they are then open to scrutiny, assessment and critique. The R package 'dagitty' takes this one step further by facilitating the evaluation of such assumptions against the implications they have for the datasets they are intended to represent. In many instances, the package can also identify adjustment sets that are unaffected by misspecification in the direction of at least some arrows within the DAG. It is therefore hoped that the R package 'dagitty' will help epidemiologists use DAGs and test them against their data with greater ease and growing confidence; and will facilitate the identification of adjustment sets that remain robust to potential misspecifications of the original DAG.

References

1. Greenland S, Pearl J, Robins JM. Causal diagrams for epidemiologic research. Epidemiol Camb Mass. 1999 Jan;10(1):37–48.
2. Rothman KJ, Greenland S, Lash TL. Modern Epidemiology. Lippincott Williams & Wilkins; 2008. 776 p.
3. Shrier I, Platt RW. Reducing bias through directed acyclic graphs. BMC Med Res Methodol. 2008;8:70.
4. Greenland S, Mansournia MA. Limitations of individual causal models, causal graphs, and ignorability assumptions, as illustrated by random confounding and design unfaithfulness. Eur J Epidemiol. 2015 Feb 17;30(10):1101–10.
5. Pearl J. Causality: Models, Reasoning and Inference. 2nd ed. New York, NY, USA: Cambridge University Press; 2009.
6. Shpitser I, VanderWeele TJ, Robins JM. On the Validity of Covariate Adjustment for Estimating Causal Effects. In: UAI 2010, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence. AUAI Press; 2010. p. 527–536.
7. Zander B van der, Liśkiewicz M, Textor J. Constructing Separators and Adjustment Sets in Ancestral Graphs. In: Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI 2014). AUAI Press; 2014. p. 907–916.
8. Perkovic E, Textor J, Kalisch M, Maathuis M. A Complete Generalized Adjustment Criterion. In: Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI 2015). AUAI Press; 2015.
9. Textor J, Hardt J, Knüppel S. DAGitty: A Graphical Tool for Analyzing Causal Diagrams. Epidemiology. 2011;5(22):745.
10. Wang L, Bautista LE. Serum bilirubin and the risk of hypertension. Int J Epidemiol. 2014 Dec 25;dyu242.
11. van Kampen D. The SSQ model of schizophrenic prodromal unfolding revised: an analysis of its causal chains based on the language of directed graphs. Eur Psychiatry J Assoc Eur Psychiatr. 2014 Sep;29(7):437–48.
12. Ooms J. V8: Embedded JavaScript Engine for R [Internet]. 2016. Available from: https://CRAN.R-project.org/package=V8
13. Shrier I. Letter to the Editor. Stat Med. 2008 Jun 30;27(14):2740–1.
14. Gansner ER, North SC. An open graph visualization system and its applications to software engineering. Softw - Pract Exp. 2000;30(11):1203–1233.
15. Shipley B. Cause and correlation in biology: a user's guide to path analysis, structural equations and causal inference. Cambridge University Press; 2002.
16. Zander B van der, Liskiewicz M. Separators and Adjustment Sets in Markov Equivalent DAGs. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. AAAI Press; 2016. p. 3315–3321.
17. Pearl J, Jewell NP, Glymour M. Causal Inference in Statistics: A Primer. Wiley; 2016.
18. Huang T-M. Testing conditional independence using maximal nonlinear conditional correlation. Ann Stat. 2010 Aug;38(4):2047–91.
19. Zhang K, Peters J, Janzing D, Schölkopf B. Kernel-based Conditional Independence Test and Application in Causal Discovery. In: UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011. AUAI Press; 2011. p. 804–813.
20. West SG, Koch T. Restoring Causal Analysis to Structural Equation ModelingReview of Causality: Models, Reasoning, and Inference (2nd Edition), by Judea Pearl. Struct Equ Model Multidiscip J. 2014 Jan 2;21(1):161–6.

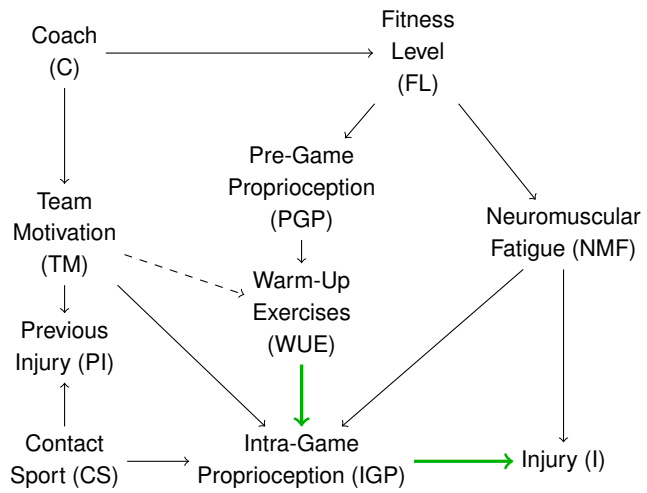
Figure Legends

Figure 1: DAG consistency evaluation. **(A)** Example DAG from a sports medicine scenario (3), lightly edited for simplicity of presentation. In this example, the DAG specified by the user is considered to be missing a causal path operating between "team motivation" (TM) and the "performance of warm-up exercises" (WUE; dashed edge). In other words, the DAG (as specified) fails to account for a relevant causal process in the application scenario. **(B)** Example code snippet showing how to load the dataset and the DAG model code (the latter pre-specified in the DAGitty web application) into an R session. The implications of the DAG are then evaluated against the dataset using a single line of R code. **(C)** Plot of the implications whose Bonferroni-Holm-corrected p-values were lower than 0.05. All three implications indicate that "team motivation" (TM) and the "performance of warm-up exercises" (WUE) must become independent when conditioned on other variables (for example, the statement "TM $\perp$ WUE | FL" means "team motivation and the performance of warm-up exercises are conditionally independent given the fitness level"). This is, however, not possible if a direct causal effect exists between the two variables (i.e. if the omission of this causal path within the DAG is, as suggested, an error), and in this instance the evaluation fails. **(D)** Adjustment sets for the original (incorrectly specified) DAG compared to those for the corrected DAG. One of the adjustment sets is minimal and valid for both DAGs; the other two adjustment sets of the corrected DAG (marked with asterisks) are valid, though not minimal, for the original DAG. **(E)** R package 'dagitty' code used to enumerate all DAGs that have the same testable implications as the DAG shown in Figure 1A (i.e. *without* the TM$\rightarrow$WUE arrow). Adjustment sets identified for all equivalent DAGs reveal that the original adjustment sets are also valid for all but one of the equivalent DAGs, in which the causal paths are changed due to several reversed directions.

Figure 2: Equivalence classes and adjustment sets. **(A)** Example DAG from a recent study exploring the potential causal relationship between bilirubin and hypertension (10), lightly edited for simplicity of presentation. **(B)** Graphical representation of the set of statistically equivalent DAGs. Bold lines indicate paths whose direction is not consistent in all of the equivalent DAGs. Two of the equivalent DAGs cannot be distinguished using the evaluation procedure illustrated in Figure 1. **(C)** Code snippet from the R package 'dagitty' illustrating how to compute an adjustment set for an entire equivalence class of DAGs. In this example, it turns out that the same minimal adjustment set is indeed valid for each of the 40 DAGs in the equivalence class. **(D)** Investigation of alternative exposure-outcome relationships for the DAG equivalence class. For each exposure-outcome combination consistent with the order of arrows in the original DAG, the number of minimal adjustment sets valid for the entire equivalence class is shown where such adjustment sets can be found.

Figure 1

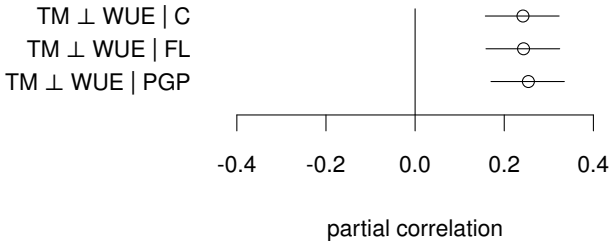
A



B

```r
# load the R package `dagitty`
library(dagitty)
# load data from a text file
d <- read.csv("http://dagitty.net/sports.csv")
# download DAG from dagitty.net
g <- downloadGraph("dagitty.net/mN4IKjR")
# evaluate the d-separation implications of the DAG
r <- localTests(g, d)
# perform Holm-Bonferroni correction
r$p.value <- p.adjust(r$p.value)
# focus on tests with p-values below a threshold
r <- r[r$p.value<0.05,]
# plot results
plotLocalTestResults(r)
```

C



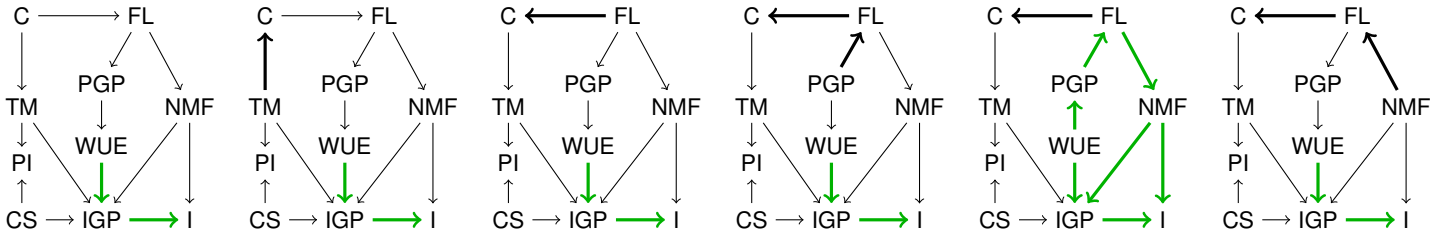
D

Minimal adjustment sets

Without TM → WUE	With TM → WUE
{ NMF, TM }	{ NMF, TM }
{ FL }	{ FL, TM }*
{ PGP }	{ PGP, TM }*
{ C, NMF }	

E

```r
# continuation of (B): enumerate the DAGs that are equivalent to (A) (without the TM->WUE arrow)
ec <- equivalentDAGs( g )
# print minimal adjustment sets for each DAG
for( i in seq_along( ec ) ){ cat(i,":\n"); print( adjustmentSets( ec[[i]], "WUE", "I" ) ) }
```



{ NMF, TM }
{ FL }
{ PGP }
{ C, NMF }

{ NMF, TM }
{ FL }
{ PGP }
{ C, NMF }

{ NMF, TM }
{ FL }
{ PGP }
{ C, NMF }

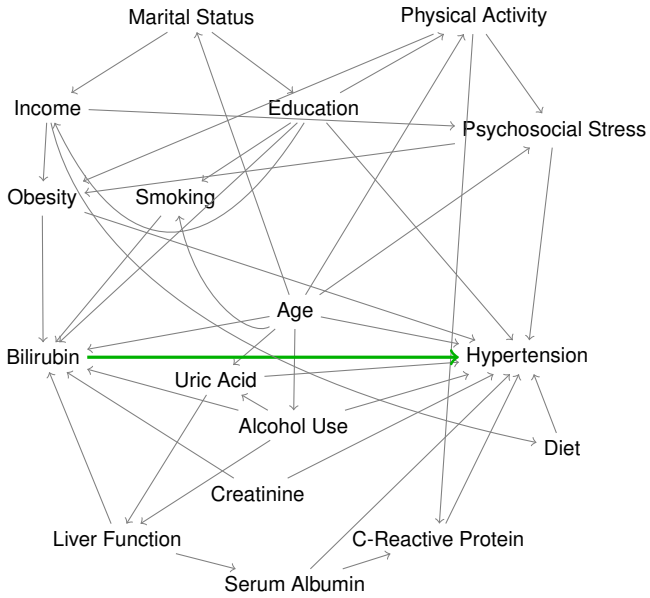
{ NMF, TM }
{ FL }
{ PGP }
{ C, NMF }

{ }

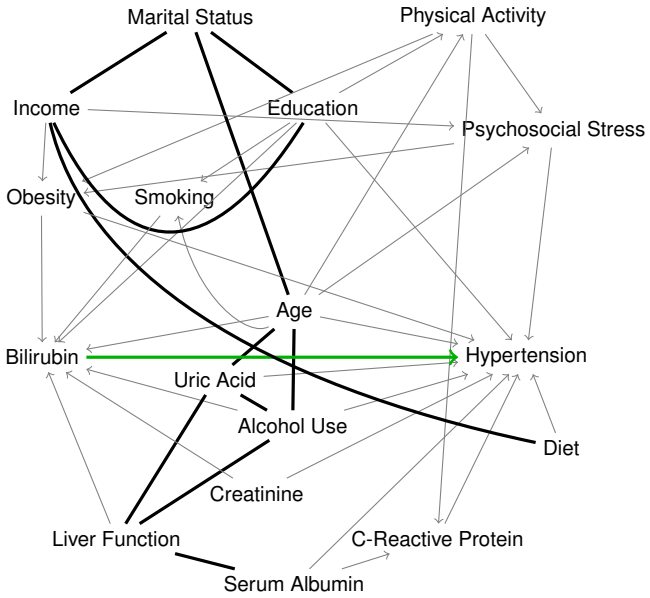
{ NMF, TM }
{ FL }
{ PGP }
{ C, NMF }

Figure 2

A



B



C

```
g <- downloadGraph("dagitty.net/mjjKjUe")
adjustmentSets(g, "Bilirubin", "Hypertension")

##  { Age, Alcohol use, Creatinine, Education, Obesity, Serum Albumin, Uric Acid }

ga <- equivalenceClass(g)
adjustmentSets(ga, "Bilirubin", "Hypertension")

##  { Age, Alcohol use, Creatinine, Education, Obesity, Serum Albumin, Uric Acid }
```

D

exposure \ outcome	Creatinine	Marital Status	Education	Smoking	Physical Activity	Income	Psychosocial Stress	Obesity	Diet	Alcohol Use	Uric Acid	Liver Function	Serum Albumin	C-Reactive Protein	Bilirubin	Hypertension
Age	1															
Creatinine		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Marital Status																
Education																
Smoking					1	1	1	1	1	1	1	1	1	1	1	1
Physical Activity									1	1	1	1	1	1		
Income																
Psychosocial Stress									1	1	1	1	1	1		
Obesity									1	1	1	1	1	1	1	4
Diet																
Alcohol Use																
Uric Acid																
Liver Function																
Serum Albumin																
C-Reactive Protein															1	3
Bilirubin																1