The Journal of Open Source Software

Causal identification with Y_0

Charles Tapley Hoyt [1*¶], **Richard J. Callahan** [2], **Joseph Cottam** [2], **August George** [2], **Benjamin M. Gyori** [3], **Haley Hummel** [4], **Nathaniel Merrill** [5], **Sara Mohammad Taheri** [3], **Pruthvi Prakash Navada** [3], **Marc-Antoine Parent** [6], **Adam Rupe** [2], **Olga Vitek** [3], **and Jeremy Zucker** [2*¶]

1 RWTH Aachen University **2** Pacific Northwest National Laboratory **3** Northeastern University **4** Oregon State University **5** Battelle Memorial Institute **6** Conversence ¶ Corresponding author * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Researchers often want to know whether one thing causes another—for example, does a new medication reduce symptoms, or does education improve income? While randomized controlled experiments provide the most direct evidence for causal relationships, they are often impossible, unethical, or prohibitively expensive to conduct for the specific questions researchers want to answer. Causal inference provides statistical methods to answer cause-and-effect questions using whatever data is available—whether observational data (collected by observing the world as it naturally occurs), experimental data from controlled studies, or a combination of both. However, determining causation is challenging because correlation does not imply causation, and many confounding factors can create misleading associations.

A key step in any causal analysis is **causal identification**—determining whether it's theoretically possible to estimate a causal effect from available data, given assumptions about relationships between variables. Causal questions exist at different levels: **interventional queries** ask "What would happen if we intervene?" (e.g., "What would be the average effect if everyone received treatment?"), while **counterfactual queries** ask "What would have happened to specific individuals in an alternative scenario?" (e.g., "Would this patient who recovered have recovered anyway without treatment?"). Modern causal identification also addresses **transportability**—determining when causal findings from one population can be validly applied to another, and how to combine evidence from multiple studies or populations to draw conclusions about a target group of interest.

The Y_0 Python package addresses a gap in the current software ecosystem by providing causal identification algorithms that handle interventional queries, counterfactual queries, and transportability challenges across different types of data. While several excellent packages exist for estimating causal effects once identification is established, Y_0 focuses specifically on the identification step—helping researchers determine *whether* a causal relationship can be estimated from their available data (observational, experimental, or mixed) before attempting to estimate *how strong* that relationship is. Y_0 provides a domain-specific language for expressing causal queries, tools for representing graphical causal models that incorporate various data types from single or multiple populations, and implementations of numerous identification algorithms from the causal inference literature.

State of the Field

Several open source packages in the Python programming language have implemented the most simple identification algorithm (`ID`) from Shpitser & Pearl (2006) including Ananke (J. J. R. Lee et al., 2023), pgmpy (Ankan & Panda, 2015), DoWhy (Sharma & Kiciman, 2020), and causaleffect-py (Pedemonte et al., 2021). Further, Ananke and DoWhy implement algorithms that consume the estimand returned by `ID` and observational data in order to estimate the average causal effect of an intervention on the outcome. However, these methods are limited in their generalization when causal queries include multiple interventions, multiple outcomes, conditionals, or interventions.

In the R programming language, the causaleffect package (Tikka & Karvanen, 2017a) implements `ID`, `IDC` (Shpitser & Pearl, 2008), surrogate outcomes (`TRSO`) (Tikka & Karvanen, 2019), and transport (S. Lee et al., 2020). The cfid package from the same authors (Tikka, 2023) implements `ID*` (Shpitser & Pearl, 2012) and `IDC*` (Shpitser & Pearl, 2012). However, these packages are challenging to use and extend.

Finally, CausalFusion is a web application that implements many identification and estimation algorithms, but is neither open source, available for registration of new users, nor provides documentation.

Causal inference remains an active research area where new identification algorithms are regularly published (see the recent review from Tikka et al. (2021)), but often without a reference implementation. This motivates the implementation of a modular framework with reusable data structures and workflows to support the implementation of both previously published and future algorithms and workflows.

Implementation

Probabilistic Expressions Y_0 implements an internal domain-specific language that can capture variables, counterfactual variables, population variables, and probabilistic expressions in which they appear. It covers the three levels of Pearl's Causal Hierarchy (Bareinboim et al., 2022), including the probability of sufficient causation $P(Y_X \mid X^*, Y^*)$, necessary causation $P(Y^*_{X^*} \mid X, Y)$, and necessary and sufficient causation $P(Y_X, Y^*_{X^*})$. Expressions can be converted to SymPy (Meurer et al., 2017), LaTeX expressions, and be rendered in Jupyter notebooks.

Data Structure Y_0 builds on NetworkX (Hagberg et al., 2008) to implement an (acyclic) directed mixed graph data structure, used in many identification algorithms, and the latent variable graph structure described by Evans (2016). It includes a suite of generic graph operations, graph simplification workflows such as the one proposed by Evans, and conversion utilities for Ananke, CausalFusion, pgmpy, and causaleffect.

Falsification Y_0 implements several workflows for checking the consistency of graphical models against observational data. First, it implements D-separation (Pearl, 2009), M-separation (Drton & Richardson, 2004), σ-separation (Forré & Mooij, 2018) that are applicable to increasingly more generic mixed graphs. Then, it implements a workflow for identifying conditional independencies (Pearl et al., 1989) and falsification (Eulig et al., 2023). Finally, it provides a wrapper around `causaleffect` through rpy2 for calculating Verma constraints (Tian & Pearl, 2012).

Identification Y_0 has the most complete suite of identification algorithms of any causal inference package. It implements `ID` (Shpitser & Pearl, 2006), `IDC` (Shpitser & Pearl, 2008), `ID*` (Shpitser & Pearl, 2012), `IDC*` (Shpitser & Pearl, 2012), surrogate outcomes (`TRSO`) (Tikka & Karvanen, 2019), `tian-ID` (Tian & Shpitser, 2010), transport (S. Lee et al., 2020), counterfactual transport (Correa et al., 2022), and identification for causal queries over hierarchical causal models (Weinstein & Blei, 2024).

The Journal of Open Source Software

Case Study

We present a case study regarding the effect of how smoking relates to cancer. First, we construct a graphical model (Figure 1A) representing the following prior knowledge:

1. Smoking causes an accumulation of tar in the lungs
2. Accumulation of tar in the lungs increase the risk of cancer
3. Smoking itself also increases the risk of cancer

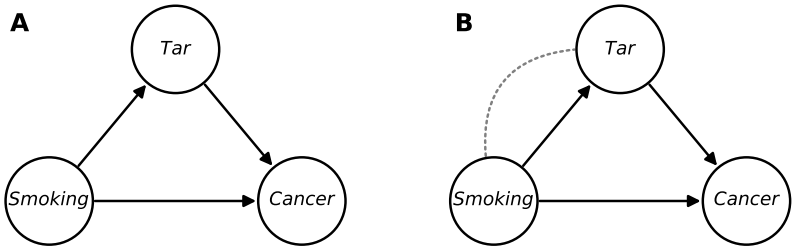

Figure 1: **A**) A simplified acyclic directed graph model representing prior knowledge on smoking and cancer and **B**) a more complex acyclic directed mixed graph that explicitly represents confounding variables.

The identification algorithm (`ID`) (Shpitser & Pearl, 2006) estimates the effect of smoking on the risk of cancer in Figure 1A as $\sum_{Tar} P(Cancer|Smoking, Tar)P(Tar|Smoking)$. However, the model in Figure 1A is inaccurate because it does not represent confounders between smoking and tar accumulation, such as the choice to smoke tar-free cigarettes. Therefore, we add a *bidirected* edge in Figure 1B. Unfortunately, `ID` can not produce an estimand for Figure 1B, which motivates the usage of an alternative algorithm that incorporates observational and/or interventional data. For example, if data from an observational study (π^*) and data from an interventional trial on smoking (π_1) are available, the surrogate outcomes algorithm (`TRSO`) (Tikka & Karvanen, 2019) estimates the effect of smoking on the risk of cancer in Figure 1B as $\sum_{Tar} P^{\pi^*}(Cancer|Smoking, Tar)P^{\pi_1}_{\text{Smoking}}(Tar)$. Code and a more detailed description of this case study can be found in the following Jupyter notebook.

We provide a second case study demonstrating the transport (S. Lee et al., 2020) and counterfactual transport (Correa et al., 2022) algorithms for epidemiological studies in COVID-19 in this Jupyter notebook.

We highlight several which used (and motivated further development of) Y_0:

- Mohammad-Taheri et al. (2022) used Y_0 to develop an automated experimental design workflow.
- Mohammad-Taheri et al. (2023) used Y_0 for falsification against experimental and simulated data for several biological signaling pathways.
- Mohammad-Taheri et al. (2024) used Y_0 and Ananke to implement an automated causal workflow for simple causal queries compatible with `ID`.
- Ness (2024) used Y_0 as a teaching tool for identification and the causal hierarchy

Future Directions

There remain several high value identification algorithms to include in Y_0 in the future. For example, the cyclic identification algorithm (`ioID`) (Forré & Mooij, 2019) is important to work with more realistic graphs that contain cycles, such as how biomolecular signaling pathways often contain feedback loops. Further, missing data identification algorithms can account for data that is missing not at random (MNAR) by modeling the underlying missingness

mechanism (Mohan & Pearl, 2021). Several algorithms noted in the review by Tikka et al. (2021), such as generalized identification (`gID`) (S. Lee et al., 2019) and generalized counterfactual identification (`gID*`) (Correa et al., 2021), can be formulated as special cases of counterfactual transportability. Therefore, we plan to improve the user experience by exposing more powerful algorithms like counterfactual transport through a simplified APIs corresponding to special cases like `gID` and `gID*`. Similarly, we plan to implement probabilistic expression simplification (Tikka & Karvanen, 2017b) to improve the consistency of the estimands output from identification algorithms.

It remains an open research question on how to estimate the causal effect for an arbitrary estimand produced by an algorithm more sophisticated than `ID`. Two potential avenues for overcoming this might be a combination of the Pyro probabilistic programming langauge (Bingham et al., 2018) and its causal inference extension ChiRho. Tractable circuits (Darwiche, 2022) also present a new paradigm for generic estimation. Such a generalization would be a lofty achievement and enable the automation of downstream applications in experimental design.

Availability and Usage

`y0` is available as a package on PyPI with the source code available at https://github.com/y0-causal-inference/y0, archived to Zenodo at doi:10.5281/zenodo.4432901, and documentation available at https://y0.readthedocs.io. The repository also contains an interactive Jupyter notebook tutorial and notebooks for the case studies described above.

Acknowledgements

The authors would like to thank the German NFDI4Chem Consortium for support. Additionally, the development of Y_0 has been partially supported by the following grants:

- DARPA award HR00111990009 (Automating Scientific Knowledge Extraction)
- PNNL Data Model Convergence Initiative award 90001 (Causal Inference and Machine Learning Methods for Analysis of Security Constrained Unit Commitment)
- DARPA award HR00112220036 (Automating Scientific Knowledge Extraction and Modeling)

The authorship of this manuscript lists the primary contributors as the first and last authors and all remaining authors in alphabetical order by family name.

References

Ankan, A., & Panda, A. (2015). *pgmpy: Probabilistic Graphical Models using Python*. 6–11. https://doi.org/10.25080/majora-7b98e3ed-001

Bareinboim, E., Correa, J. D., Ibeling, D., & Icard, T. (2022). On pearl's hierarchy and the foundations of causal inference. In *Probabilistic and causal inference: The works of judea pearl* (1st ed., pp. 507–556). Association for Computing Machinery. https://doi.org/10.1145/3501714.3501743

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., & Goodman, N. D. (2018). *Pyro: Deep universal probabilistic programming*. https://arxiv.org/abs/1810.09538

Correa, J. D., Lee, S., & Bareinboim, E. (2022). Counterfactual transportability: A formal approach. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, & S. Sabato (Eds.), *Proceedings of the 39th international conference on machine learning* (Vol. 162, pp. 4370–4390). PMLR. https://proceedings.mlr.press/v162/correa22a.html

Correa, J. D., Lee, S., & Bareinboim, E. (2021). Nested counterfactual identification from arbitrary surrogate experiments. *Advances in Neural Information Processing Systems*, *34*. https://arxiv.org/abs/2107.03190

Darwiche, A. (2022). *Causal inference using tractable circuits*. https://arxiv.org/abs/2202.02891

Drton, M., & Richardson, T. S. (2004). Iterative conditional fitting for gaussian ancestral graph models. *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 130–137. ISBN: 0974903906

Eulig, E., Mastakouri, A. A., Blöbaum, P., Hardt, M., & Janzing, D. (2023). *Toward falsifying causal graphs using a permutation-based test*. https://arxiv.org/abs/2305.09565

Evans, R. J. (2016). Graphs for margins of bayesian networks. *Scandinavian Journal of Statistics*, *43*(3), 625–648. https://doi.org/10.1111/sjos.12194

Forré, P., & Mooij, J. M. (2018). *Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders*. https://arxiv.org/abs/1807.03024

Forré, P., & Mooij, J. M. (2019). *Causal calculus in the presence of cycles, latent confounders and selection bias*. https://arxiv.org/abs/1901.00433

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15). https://aric.hagberg.org/papers/hagberg-2008-exploring.pdf

Lee, J. J. R., Bhattacharya, R., Nabi, R., & Shpitser, I. (2023). *Ananke: A python package for causal inference using graphical models*. https://arxiv.org/abs/2301.11477

Lee, S., Correa, J. D., & Bareinboim, E. (2019). General identifiability with arbitrary surrogate experiments. *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*. https://proceedings.mlr.press/v115/lee20b.html

Lee, S., Correa, J. D., & Bareinboim, E. (2020). General transportability – synthesizing observations and experiments from heterogeneous domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(06), 10210–10217. https://doi.org/10.1609/aaai.v34i06.6582

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., … Scopatz, A. (2017). SymPy: Symbolic computing in python. *PeerJ Computer Science*, *3*, e103. https://doi.org/10.7717/peerj-cs.103

Mohammad-Taheri, S., Navada, P. P., Hoyt, C. T., Zucker, J., Sachs, K., Gyori, B., & Vitek, O. (2024). Eliater: A python package for estimating outcomes of perturbations in biomolecular networks. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btae527

Mohammad-Taheri, S., Tewari, V., Kapre, R., Rahiminasab, E., Sachs, K., Hoyt, C. T., Zucker, J., & Vitek, O. (2022). *Experimental design for causal query estimation in partially observed biomolecular networks*. https://arxiv.org/abs/2210.13423

Mohammad-Taheri, S., Tewari, V., Kapre, R., Rahiminasab, E., Sachs, K., Hoyt, C. T., Zucker, J., & Vitek, O. (2023). Optimal adjustment sets for causal query estimation in partially observed biomolecular networks. *Bioinformatics*, *39*(Supplement_1), i494–i503. https://doi.org/10.1093/bioinformatics/btad270

Mohan, K., & Pearl, J. (2021). Graphical models for processing missing data. *Journal of the American Statistical Association*, *116*(534), 1023–1037. https://doi.org/10.1080/01621459.2021.1874961

Ness, R. (2024). *Causal AI*. O'Reilly Media. ISBN: 978-1-63343-991-7

Pearl, J. (2009). *Causality* (2nd ed.). Cambridge University Press. ISBN: 978-0521895606

Pearl, J., Geiger, D., & Verma, T. S. (1989). Conditional independence and its representations. *Kybernetika*, *25*(Suppl), 33–44. http://eudml.org/doc/28568

Pedemonte, M., Vitrià, J., & Parafita, Á. (2021). *Algorithmic causal effect identification with causaleffect*. https://arxiv.org/abs/2107.04632

Sharma, A., & Kiciman, E. (2020). *DoWhy: An end-to-end library for causal inference*. https://arxiv.org/abs/2011.04216

Shpitser, I., & Pearl, J. (2006). Identification of joint interventional distributions in recursive semi-markovian causal models. *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, 1219–1226. https://doi.org/10.5555/1597348.1597382

Shpitser, I., & Pearl, J. (2008). Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research*, *9*(64), 1941–1979. http://jmlr.org/papers/v9/shpitser08a.html

Shpitser, I., & Pearl, J. (2012). *What counterfactuals can be tested*. https://arxiv.org/abs/1206.5294

Tian, J., & Pearl, J. (2012). *On the testable implications of causal models with hidden variables*. https://arxiv.org/abs/1301.0608

Tian, J., & Shpitser, I. (2010). On identifying causal effects. *Heuristics, Probability and Causality: A Tribute to Judea Pearl (R. Dechter, H. Geffner and J. Halpern, Eds.). College Publications, UK*, 415–444. https://faculty.sites.iastate.edu/jtian/files/inline-files/tian-shpitser-2009.pdf

Tikka, S. (2023). Identifying counterfactual queries with the r package cfid. *The R Journal*, *15*, 330–343. https://doi.org/10.32614/RJ-2023-053

Tikka, S., Hyttinen, A., & Karvanen, J. (2021). Causal effect identification from multiple incomplete data sources: A general search-based approach. *Journal of Statistical Software*, *99*(5), 1–40. https://doi.org/10.18637/jss.v099.i05

Tikka, S., & Karvanen, J. (2017a). Identifying causal effects with the r package causaleffect. *Journal of Statistical Software*, *76*(12), 1–30. https://doi.org/10.18637/jss.v076.i12

Tikka, S., & Karvanen, J. (2017b). Simplifying probabilistic expressions in causal inference. *Journal of Machine Learning Research*, *18*(36), 1–30. http://jmlr.org/papers/v18/16-166.html

Tikka, S., & Karvanen, J. (2019). Surrogate outcomes and transportability. *International Journal of Approximate Reasoning*, *108*, 21–37. https://doi.org/10.1016/j.ijar.2019.02.007

Weinstein, E. N., & Blei, D. M. (2024). *Hierarchical causal models*. https://arxiv.org/abs/2401.05330