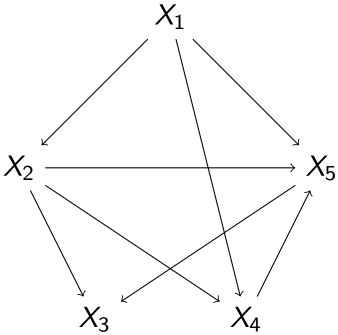
UNIVERSITY OF COPENHAGEN

Are You Doing Better Than Random Guessing? A Call for Using Negative Controls When Evaluating Causal Discovery Algorithms

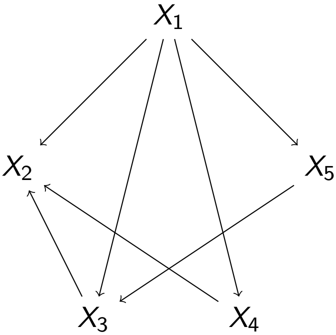
Anne Helby Petersen

Example: Comparing two DAGs

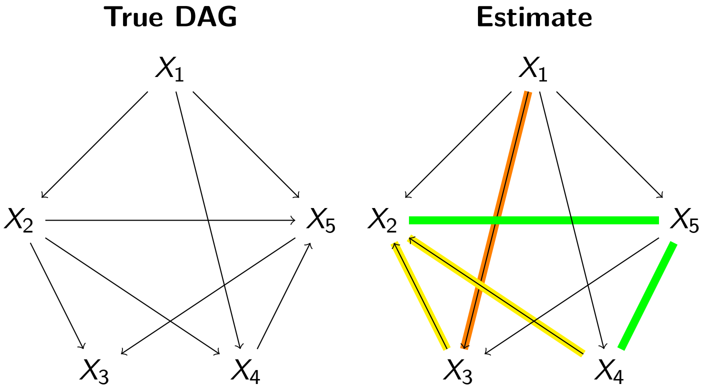
True DAG



Estimate

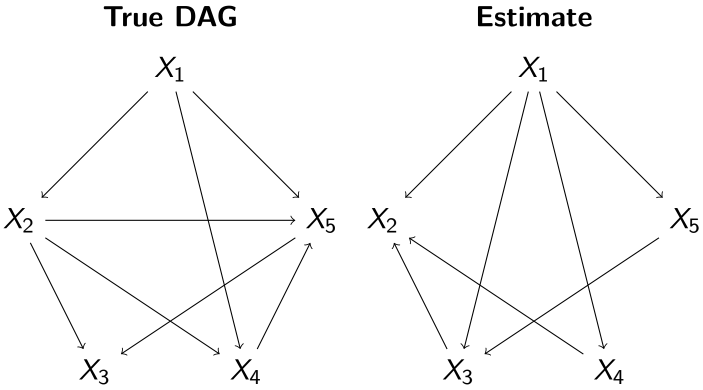


Example: Comparing two DAGs



Structural hamming distance: 5

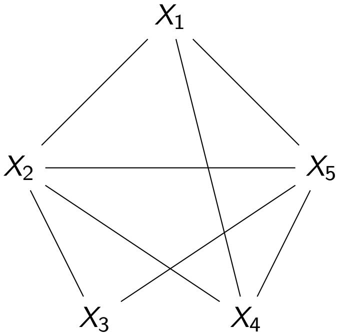
Example: Comparing two DAGs



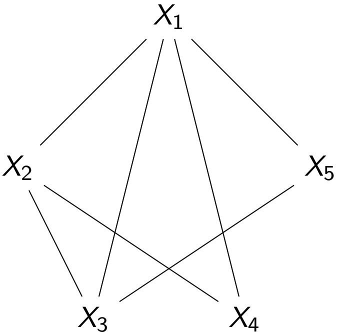
Structural hamming distance: 5

Example: Comparing two DAGs

True DAG



Estimate

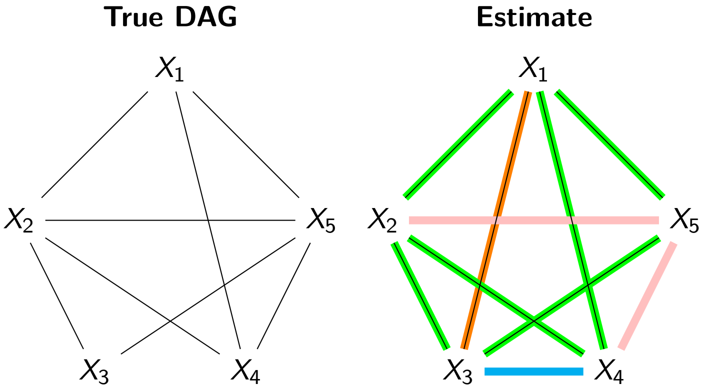


Structural hamming distance: 5

Adjacency precision: $\frac{TP}{TP+FP}$

Adjacency recall: $\frac{TP}{TP+FN}$

Example: Comparing two DAGs



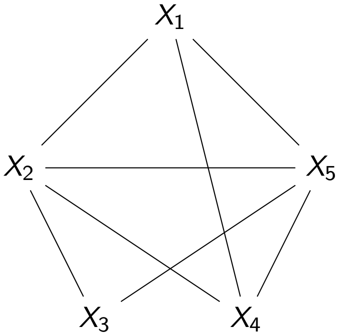
Structural hamming distance: 5

Skeleton estimation: TP: 6, FP: 1 , TN: 1, FN: 2

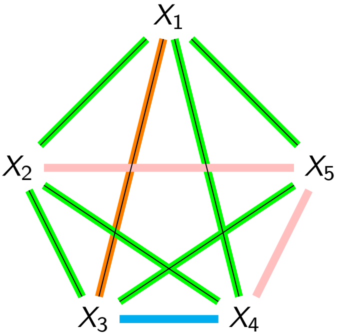
Adjacency precision: $\frac{TP}{TP+FP}$ **Adjacency recall**: $\frac{TP}{TP+FN}$

Example: Comparing two DAGs

True DAG



Estimate

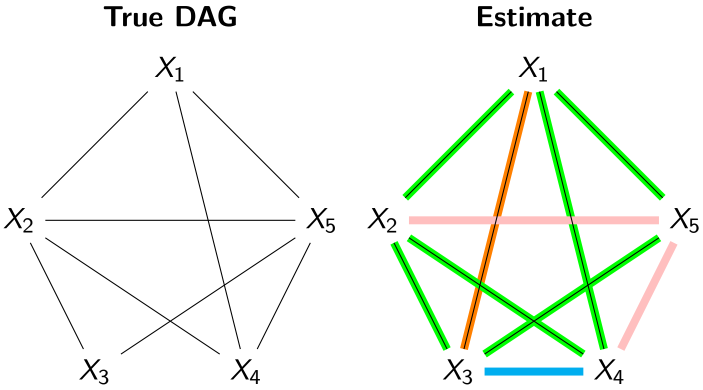


Structural hamming distance: 5

Skeleton estimation: TP: 6, FP: 1 , TN: 1, FN: 2

Adjacency precision: $\frac{TP}{TP+FP} \simeq 0.86$ **Adjacency recall**: $\frac{TP}{TP+FN} = 0.75$

Example: Comparing two DAGs



Structural hamming distance: 5

Skeleton estimation: TP: 6, FP: 1 , TN: 1, FN: 2

Adjacency precision: $\frac{TP}{TP+FP} \simeq 0.86$ **Adjacency recall**: $\frac{TP}{TP+FN} = 0.75$

Are these numbers **big or small**? **Good or bad CD algorithm?**

A random guessing baseline

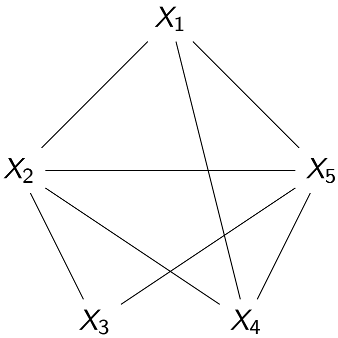
Idea: Use **random guessing** as a simple common baseline for evaluating causal discovery algorithms

- Makes it easier to determine which causal discovery problems are **"easy"** and which ones are **"hard"**
- Increases **interpretability** of reported metrics across different simulation study designs
- Describes how **informative** a given evaluation study/metric is (can high performance be attained trivially?)

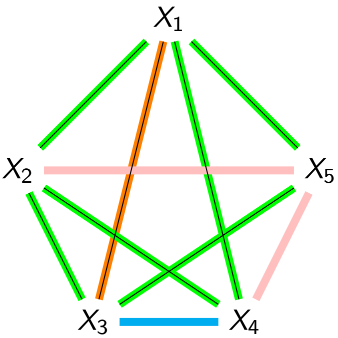
This can be viewed as a *negative control* concept – current evaluations are only using *positive controls* (i.e. other causal discovery algorithm) for comparisons.

Skeleton estimation under random guessing

True DAG



Estimate



		Truth		
		Adjacency	Non-adjacency	Total
Estimate	Adjacency	TP	FP	m_{est}
	Non-adjacency	FN	TN	-
	Total	m_{true}	-	$m_{\max}$

Skeleton estimation under random guessing

		Truth: Adjacency	Truth: Non-adjacency	Total
Estimate	Adjacency	TP	FP	m_{est}
	Non-adjacency	FN	TN	-
	Total	m_{true}	-	m_{max}

- $\boldsymbol{m}_{\text{max}} = \frac{1}{2}(d-1)d$ is a mathematical property of the graph.
- If there exists a known ground truth, $\boldsymbol{m}_{\text{true}}$ is fixed and known.
- For standard applications of most CD algorithms, $\boldsymbol{m}_{\text{est}}$ is *not* estimated, but chosen indirectly via e.g. a test significance level set at e.g. 0.05. So we also consider m_{est} fixed *(if not ok: revert to simulation-based approach, shown later)*.

Skeleton estimation under random guessing

		Truth: Adjacency	Non-adjacency	Total
Estimate	Adjacency	TP	FP	m_{est}
	Non-adjacency	FN	TN	-
	Total	m_{true}	-	$m_{\max}$

- $\boldsymbol{m}_{\max} = \frac{1}{2}(d-1)d$ is a mathematical property of the graph.
- If there exists a known ground truth, $\boldsymbol{m}_{\text{true}}$ is fixed and known.
- For standard applications of most CD algorithms, $\boldsymbol{m}_{\text{est}}$ is *not* estimated, but chosen indirectly via e.g. a test significance level set at e.g. 0.05. So we also consider m_{est} fixed *(if not ok: revert to simulation-based approach, shown later)*.

Key observation: Under random edge placement in the estimated graph, we have that

$$TP \mid m_{\max}, m_{\text{true}}, m_{\text{est}} \sim \text{HyperGeom}(m_{\max}, m_{\text{true}}, m_{\text{est}})$$

Note: Exact distributional result! (Same as used for Fisher's exact test...)

Two usecases for inference about skeleton estimation

1: Expectations + CIs for ML metrics under random guessing

Metric	Expected value	Quantile
Precision	$\frac{m_{\text{true}}}{m_{\text{max}}}$	$\frac{k_q}{m_{\text{est}}}$
Recall	$\frac{m_{\text{est}}}{m_{\text{max}}}$	$\frac{k_q}{m_{\text{true}}}$
F1	$\frac{2 \cdot m_{\text{est}} \cdot m_{\text{true}}}{m_{\text{max}} \cdot m_{\text{est}} + m_{\text{max}} \cdot m_{\text{true}}}$	$\frac{2 \cdot k_q}{m_{\text{est}} + m_{\text{true}}}$
...		

Two usecases for inference about skeleton estimation

1: Expectations + CIs for ML metrics under random guessing

Metric	Expected value	Quantile
Precision	$\frac{m_{\text{true}}}{m_{\text{max}}}$	$\frac{k_q}{m_{\text{est}}}$
Recall	$\frac{m_{\text{est}}}{m_{\text{max}}}$	$\frac{k_q}{m_{\text{true}}}$
F1	$\frac{2 \cdot m_{\text{est}} \cdot m_{\text{true}}}{m_{\text{max}} \cdot m_{\text{est}} + m_{\text{max}} \cdot m_{\text{true}}}$	$\frac{2 \cdot k_q}{m_{\text{est}} + m_{\text{true}}}$
...		

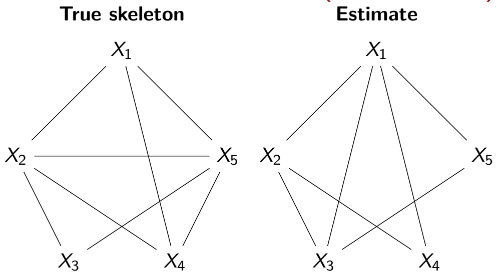
2: Overall test of skeleton fit

Let G be the true graph with m_{true} edges and $\hat{G}$ be an estimated graph with m_{est} edges. We can then conduct an **exact test** of

$$H_0 : \hat{G} \text{ was obtained by randomly placing } m_{\text{est}} \text{ edges.}$$

by computing a one-sided **p-value** as $p = P(X \geq TP_{obs})$ where $X \sim \text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$.

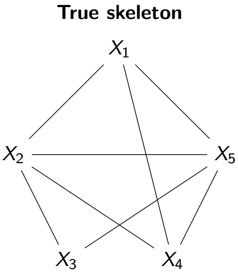
Example: Two DAG skeletons (continued)

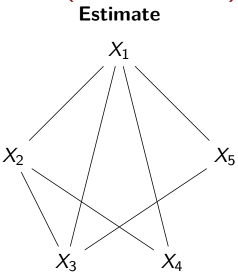


Precision: $\frac{TP}{TP+FP} \simeq 0.86$. **Recall**: $\frac{TP}{TP+FN} = 0.75$

Are these numbers **big or small**? **Good or bad CD algorithm?**

Example: Two DAG skeletons (continued)



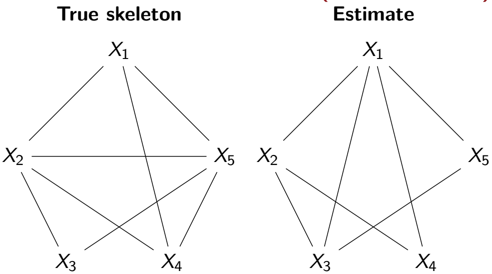


Precision: $\frac{TP}{TP+FP} \simeq 0.86$. **Recall**: $\frac{TP}{TP+FN} = 0.75$

Are these numbers **big or small**? **Good or bad CD algorithm?**

- Negative control expected **precision**: $\frac{m_{\text{true}}}{m_{\text{max}}} = 0.80 \quad (0.71, 1.00)$

Example: Two DAG skeletons (continued)

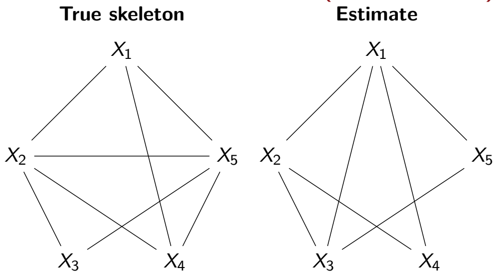


Precision: $\frac{TP}{TP+FP} \simeq 0.86$. **Recall**: $\frac{TP}{TP+FN} = 0.75$

Are these numbers **big or small**? **Good or bad CD algorithm?**

- Negative control expected **precision**: $\frac{m_{\text{true}}}{m_{\text{max}}} = 0.80 \quad (0.71, 1.00)$
- Negative control expected **recall**: $\frac{m_{\text{est}}}{m_{\text{max}}} = 0.70 \quad (0.63, 0.88)$

Example: Two DAG skeletons (continued)



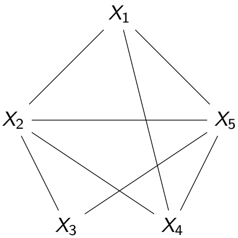
Precision: $\frac{TP}{TP+FP} \simeq 0.86$. **Recall**: $\frac{TP}{TP+FN} = 0.75$

Are these numbers **big or small**? **Good or bad CD algorithm?**

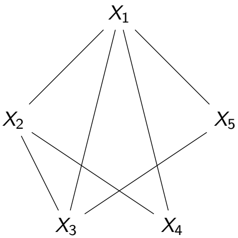
- Negative control expected **precision**: $\frac{m_{\text{true}}}{m_{\text{max}}} = 0.80 \quad (0.71, 1.00)$
- Negative control expected **recall**: $\frac{m_{\text{est}}}{m_{\text{max}}} = 0.70 \quad (0.63, 0.88)$
- Test of overall skeleton fit: $p = 0.53$.

Example: Two DAG skeletons (continued)

True skeleton



Estimate

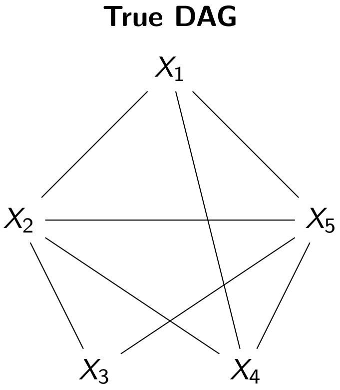


Precision: $\frac{TP}{TP+FP} \simeq 0.86$. **Recall**: $\frac{TP}{TP+FN} = 0.75$

Are these numbers **big or small**? **Good or bad CD algorithm?**

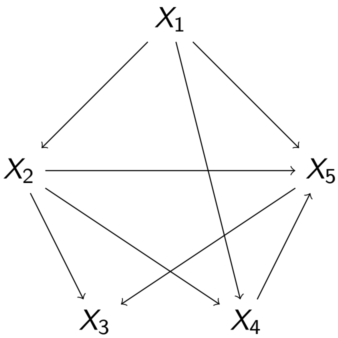
- Negative control expected **precision**: $\frac{m_{\text{true}}}{m_{\max}} = 0.80 \quad (0.71, 1.00)$
- Negative control expected **recall**: $\frac{m_{\text{est}}}{m_{\max}} = 0.70 \quad (0.63, 0.88)$
- Test of overall skeleton fit: $p = 0.53$.
- So **not impressive** causal discovery (... because it *was* random guessing)

Beyond adjacencies: Bringing back orientations

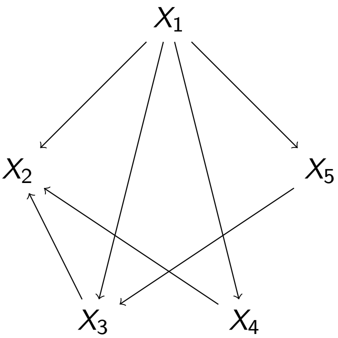


Beyond adjacencies: Bringing back orientations

True DAG



Estimate



Beyond adjacencies: Simulation-based pipeline

1. **Standard simulation study:** Conduct simulation study as usual, compute metric of interest for each simulated graph + store the true (simulated) DAG + m_{est}.
2. **Negative control simulation:** Draw a large (e.g., 1000) number of random DAGs with number of edges sampled from the m_{est} distribution from Step 1. Compute metric of interest for each neg. control.
3. **Comparison:** Conduct statistical inference on pairwise differences in metric from Step 1 (causal discovery) vs. Step 2 (negative control). Report with p-values/confidence intervals from empirical distribution.

Example: PC algorithm evaluation (1/3)

Simulation study: 1000 random DAGs ($d = 10$ nodes) + linear Gaussian data ($n = 1000$). Nice case for PC: Will find true CPDAG in large sample limit.

Two simulation settings:

1. **Dense graphs** ($m_{\text{true}} = 30$). PC algorithm on finite data is biased towards sparse graphs [1] $\Rightarrow$ struggles on dense graphs. **Expectation: No difference between neg. control and PC.**
2. **Sparser graphs** ($m_{\text{true}} = 15$). Easier case for PC. **Expectation: PC better than neg. control.**

Report: Means and 95% CIs for each metric + one-sided p-value for pairwise differences in metrics (PC vs. neg. control).

[1] Petersen, Ramsey, Ekstrøm, & Spirtes (2022). Causal discovery for observational sciences using supervised machine learning. Journal of Data Science.

Example: PC algorithm evaluation (2/3)

Case 1: Dense graphs ($m_{\text{true}} = 30$). **PC known to struggle.**

	PC		Negative control		
	Mean	CI	Mean	CI	p
SHD	27.33	(21,33)	31.23	(26, 36)	0.202
Adjacency precision	0.85	(0.65, 1.00)	0.66	(0.42, 0.87)	0.122
Adjacency recall	0.38	(0.27, 0.50)	0.29	(0.17, 0.43)	0.245
Orientation precision	0.65	(0, 1)	0.50	(0, 1)	0.360
Orientation recall	0.40	(0.00, 0.78)	0.37	(0.00, 0.78)	0.464
Recovered v-struct.	0.05	(0.0, 0.2)	0.02	(0.00, 0.14)	0.563
SID (lower bound)	67.73	(46, 83)	74.23	(56, 85)	0.317
SID (upper bound)	79.48	(61, 90)	79.10	(63, 88)	0.557

Example: PC algorithm evaluation (2/3)

Case 1: Dense graphs ($m_{\text{true}} = 30$). **PC known to struggle.**

	PC		Negative control		
	Mean	CI	Mean	CI	p
SHD	27.33	(21,33)	31.23	(26, 36)	0.202
Adjacency precision	0.85	(0.65, 1.00)	0.66	(0.42, 0.87)	0.122
Adjacency recall	0.38	(0.27, 0.50)	0.29	(0.17, 0.43)	0.245
Orientation precision	0.65	(0, 1)	0.50	(0, 1)	0.360
Orientation recall	0.40	(0.00, 0.78)	0.37	(0.00, 0.78)	0.464
Recovered v-struct.	0.05	(0.0, 0.2)	0.02	(0.00, 0.14)	0.563
SID (lower bound)	67.73	(46, 83)	74.23	(56, 85)	0.317
SID (upper bound)	79.48	(61, 90)	79.10	(63, 88)	0.557

Results as expected: **No significant differences between PC and neg. control** (at e.g. 10% level).

Example: PC algorithm evaluation (3/3)

Case 2: Sparser graphs ($m_{\text{true}} = 15$). **PC known to work well.**

	PC		Negative control		
	Mean	CI	Mean	CI	p
SHD	10.1	$(4, 15)$	21.30	$(17, 25)$	0.002
Adjacency precision	0.9	$(0.73, 1.00)$	0.33	$(0.09, 0.57)$	0.000
Adjacency recall	0.7	$(0.47, 0.87)$	0.25	$(0.07, 0.47)$	0.001
Orientation precision	0.9	$(0.5, 1.0)$	0.52	$(0, 1)$	0.273
Orientation recall	0.5	$(0.00, 0.91)$	0.36	$(0, 1)$	0.316
Recovered v-struct.	0.3	$(0.0, 0.8)$	0.01	$(0.00, 0.14)$	0.106
SID (lower bound)	29.3	$(7, 55)$	51.01	$(29, 74)$	0.072
SID (upper bound)	51.5	$(22, 81)$	58.43	$(36, 81)$	0.350

Example: PC algorithm evaluation (3/3)

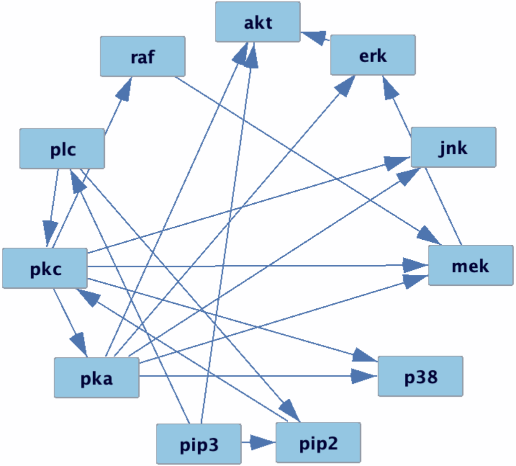
Case 2: Sparser graphs ($m_{\text{true}} = 15$). **PC known to work well.**

	PC		Negative control		
	Mean	CI	Mean	CI	p
SHD	10.1	$(4, 15)$	21.30	$(17, 25)$	0.002
Adjacency precision	0.9	$(0.73, 1.00)$	0.33	$(0.09, 0.57)$	0.000
Adjacency recall	0.7	$(0.47, 0.87)$	0.25	$(0.07, 0.47)$	0.001
Orientation precision	0.9	$(0.5, 1.0)$	0.52	$(0, 1)$	0.273
Orientation recall	0.5	$(0.00, 0.91)$	0.36	$(0, 1)$	0.316
Recovered v-struct.	0.3	$(0.0, 0.8)$	0.01	$(0.00, 0.14)$	0.106
SID (lower bound)	29.3	$(7, 55)$	51.01	$(29, 74)$	0.072
SID (upper bound)	51.5	$(22, 81)$	58.43	$(36, 81)$	0.350

Some metrics are able to pick up difference between PC and neg. control, but not all. May suggest some metrics are non-informative for this task. . .

Example: Sachs data

Sachs dataset: Commonly used benchmark dataset on protein signaling with ground truth DAG with 11 nodes, $m_{\text{true}} = 20$ edges.



SHD on Sachs dataset ($m_{\text{true}} = 20$)

	Observed	
	SHD	m_{est}
NOTEARS	22	16
PC	23	24
BOSS	35	32
LiNGAM	30	33
GES	30	30

SHD on Sachs dataset ($m_{\text{true}} = 20$)

	Observed		Negative controls	
	SHD	m_{est}	$\overline{\text{SHD}}$	p
NOTEARS	22	16	27.1	0.050
PC	23	24	31.5	0.001
BOSS	35	32	35.2	0.510
LiNGAM	30	33	34.4	0.083
GES	30	30	34.2	0.114

Simulation-based negative controls:

- **Negative controls**: Draw 1000 random DAGs (Erdös-Rényi) over 11 nodes with m_{est} edges (seperately for each m_{est}). Compare each with Sachs ground truth, compute SHD, report mean.
- One-sided **p-values** testing H_0 : Neg. control at least as good as algorithm. Computed from empirical neg. control SHD distributions.

Conclusions

Interpreting causal discovery evaluations – even with known ground truth – is not as simple as may seem:

- We're often comparing **apples and oranges**: Applying commonly used metrics such as SHD/precision/recall across graphs with different estimated or true sparsities is **not meaningful**.
- **Not all metrics are informative** for all discovery tasks/all choices of true sparsities (m_{true}) and estimated sparsities (m_{est}).
- **Negative control baseline helps a lot!**
- Negative controls are **simple to do**, and for the widely used skeleton metrics, we provide closed formulas for expected values etc. **Code**: https://github.com/annennenne/negcontrol-disco

All this should of course be supplemented with **real data applications** to assess if causal discovery provides **useful and novel information in practice.**

Thank you!