PC Algorithm Results Report

ENLA Pipeline

October 19, 2025

1 Overview

This report documents the causal discovery analyses performed with the Peter–Clark (PC) algorithm within the ENLA pipeline. The goal is to recover directed causal structures among questionnaire indicators and key academic outcomes:

- socioemotional learning composite (`socioemotional_learning`)
- academic language performance (`academic_lang`)
- academic mathematics performance (`academic_math`)

All experiments rely on the preprocessed dataset generated by `pipeline/02_preprocess.R` and are executed via `pipeline/05_pc_algorithm.R`.

2 Methodology

2.1 Data Preparation

We use the preprocessed object at `outputs/preprocessed/enla_preprocessed.rds`. The `analysis` component contains harmonized performance indicators and weights; the `raw` component exposes the questionnaire microdata. We select the student questionnaire (`PC_QUESTIONNAIRE=estudiante`) and align records through `id_estudiante`. Questionnaire responses are restricted to numeric variables, then merged with `academic_lang`, `academic_math`, and `socioemotional_learning`. Prior to estimation, all variables are standardized (z-scores) to remove scale effects and improve numerical conditioning.

2.2 PC Algorithm Configuration

The PC algorithm is fit at significance level $\alpha = 0.005$. In this report we present results at maximum conditioning depth $m = 4$ for both the questions and latent networks. The stable skeleton search is employed; orientations are derived from v-structures and Meek's rules, and we retain undirected edges where directionality is not identified. For the present results, we disabled bootstrapping to prioritize runtime and reproducibility.

2.3 Key Hyperparameters and Statistical Tests

The PC algorithm is a constraint-based method that learns a graph by testing conditional independences and then orienting edges under causal sufficiency assumptions. We summarize the main control parameters and tests used in this report.

$m_{\max}$ **(maximum conditioning set size).** During skeleton discovery, the algorithm evaluates hypotheses of the form $X \perp Y \mid S$ for conditioning sets $S \subseteq V \setminus \{X, Y\}$ whose cardinality does not exceed $m_{\max}$. Increasing $m_{\max}$ permits conditioning on larger sets, which can further remove edges that become independent only after controlling for more variables, but the number of tests grows combinatorially with $|S|$, increasing runtime and variance of the decisions.

α **(test size / significance level).** Each conditional independence is assessed at level α. If the p-value exceeds α, the null of independence is not rejected and the edge X–Y is removed; otherwise the edge is retained. Smaller α yields sparser graphs (fewer false positives at the risk of more false negatives), while larger α yields denser graphs.

Independence test (`gaussCItest`). With standardized variables and sample size n, the Gaussian test uses the Fisher z-transform of the sample partial correlation $r_{XY \cdot S}$, with statistic

$$z = \tfrac{1}{2} \log\left(\frac{1 + r_{XY \cdot S}}{1 - r_{XY \cdot S}}\right) \sqrt{n - |S| - 3}\,,$$

which is asymptotically standard normal under $H_0 \colon \rho_{XY \cdot S} = 0$. The stable skeleton variant is used to make the result order independent; orientations are then obtained by identifying v-structures and propagating arrowheads via Meek's rules.

2.4 Survey Weights and Design-Based Considerations

Large-scale education surveys routinely provide sampling or nonresponse weights to recover population-representative estimands. In the PC workflow driven by conditional independence tests on partial correlations, weights can be incorporated through a design-consistent plug-in of first- and second-order moments, together with an effective sample size.

Weighted moments and correlations. Let $w_i > 0$ denote the analysis weight for observation $i = 1, \ldots, n$. For a variable vector $X \in \mathbb{R}^p$, define the weighted mean and covariance as

$$\mu_w = \frac{\sum_{i=1}^n w_i X_i}{\sum_{i=1}^n w_i}, \qquad \Sigma_w = \frac{\sum_{i=1}^n w_i (X_i - \mu_w)(X_i - \mu_w)^\top}{\sum_{i=1}^n w_i}.$$

The weighted correlation matrix C_w is obtained by standardizing Σ_w by its diagonal. Supplying *weighted* sufficiency statistics to the Gaussian CI test is then natural: `suffStat = {C =` C_w`, n =` n_{eff}`}`.

Effective sample size. Because unequal weights inflate variance relative to simple random sampling, we replace n with the Kish effective sample size

$$n_{\text{eff}} = \frac{\left(\sum_{i=1}^n w_i\right)^2}{\sum_{i=1}^n w_i^2},$$

which downscales the test's degrees of freedom and yields more conservative decisions when weights are highly variable. In practice, this tends to reduce spurious edges and aligns CI testing with the intended target population.

Replicate-weight stability (optional). If replicate weights (e.g., BRR/JK/bootstrap) are available, a design-based robustness check is to run PC separately for each replicate $r = 1, \ldots, R$ and summarize edges by inclusion frequency

$$\pi_e = \frac{1}{R} \sum_{r=1}^{R} \mathbb{1}\{e \in E^{(r)}\},$$

reporting only edges with π_e above a threshold (and orienting edges when arrowheads are stable). This complements the weighted-moment plug-in by capturing design-induced variability.

Scope and limitations. The weighted plug-in (C_w, n_{eff}) respects population representation and integrates seamlessly with the Gaussian CI test; however, it does not by itself model clustering or stratification beyond their effect on weights. When complex design features are substantial, replicate-weight stability is recommended for inference on edge presence and orientation.

2.5 Network Variants

We build two complementary networks per questionnaire. The *questions network* uses raw item responses (e.g., `pXX`, `pXX_YY`), capturing granular behaviors and perceptions. The *latent network* uses composite scales (e.g., `EST6*`, `EST6P*`), summarizing broader constructs. Both networks include the three target outcomes: `academic_lang`, `academic_math`, and `socioemotional_learning`.

3 Runtime Summary

Table 1 consolidates all recorded runtimes stored in `outputs/pc_algorithm/pc_run_*.rds`.

Table 1: PC algorithm runtime summary

File	Questionnaire	Network	α	$m_{\max}$	Runtime (s)	Edges	Nodes
pc_run_estudiante_questions_m3.rds	estudiante	questions (m=3)	0.005	3	74.24	383	40
pc_run_estudiante_latent_m3.rds	estudiante	latent (m=3)	0.005	3	7.47	135	28
pc_run_estudiante_questions_m4.rds	estudiante	questions (m=4)	0.005	4	216.87	338	40
pc_run_estudiante_latent_m4.rds	estudiante	latent (m=4)	0.005	4	14.52	118	28

The initial combined runs (prior to splitting) captured only the top-variance numeric fields from the analysis dataset; while informative, later analyses focus on questionnaire-specific subsets.

4 Causal Parents of Key Outcomes

We identify drivers of the three outcomes (`academic_lang`, `academic_math`, `socioemotional_learning`) by extracting directed parents from the learned graphs (undirected ties are excluded). Under the settings reported here, the questions network at $m = 4$ shows no directed parents into the three outcomes at $\alpha = 0.005$. In contrast, the latent network at $m = 4$ reveals two directed parents into `socioemotional_learning`: `est6pgen_estestsi` and `est6pgen_opparest`. A consolidated summary (including $m = 3$ for comparison) is provided in Table 2.

5 Network Characteristics

5.1 Depth-4 Networks

At depth $m = 4$, the PC algorithm explores even larger conditioning sets, further pruning edges and orienting structures when supported by the data. The student questions network has 338 edges over 40 nodes, while the latent network has 118 edges over 28 nodes.

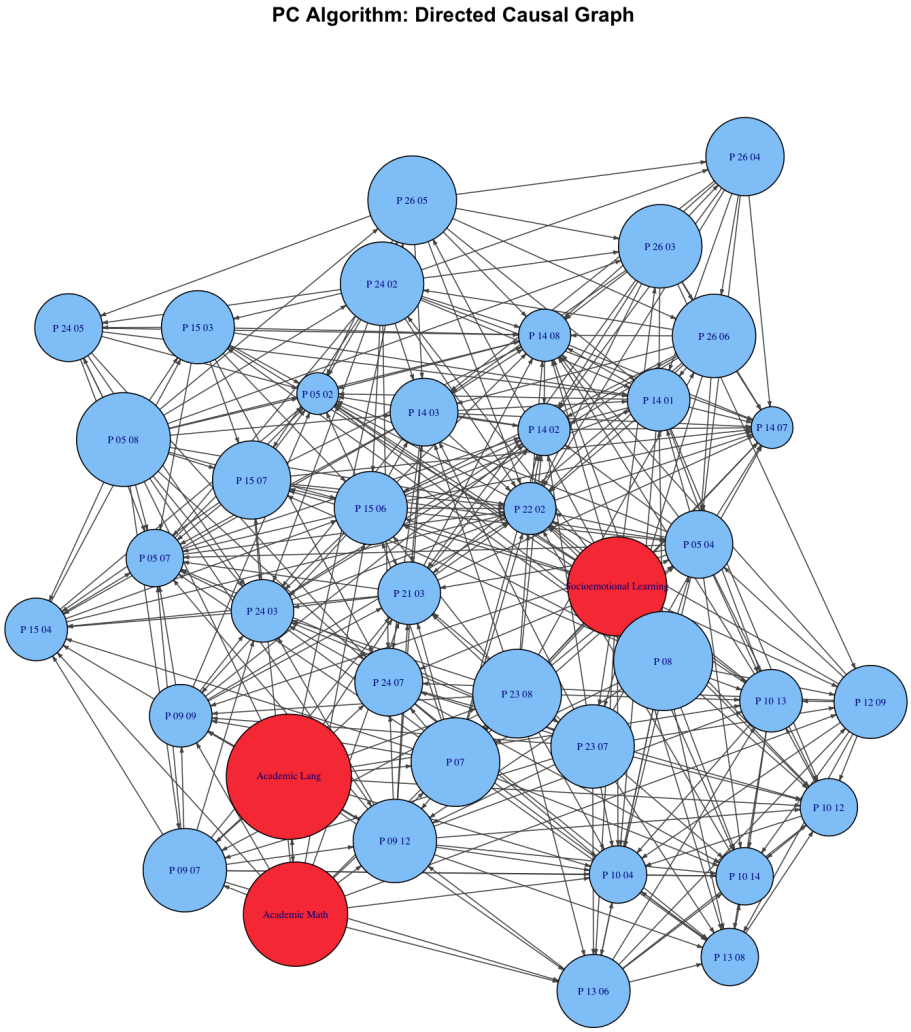


Figure 1: Questions network (m=4): full graph (student questionnaire).

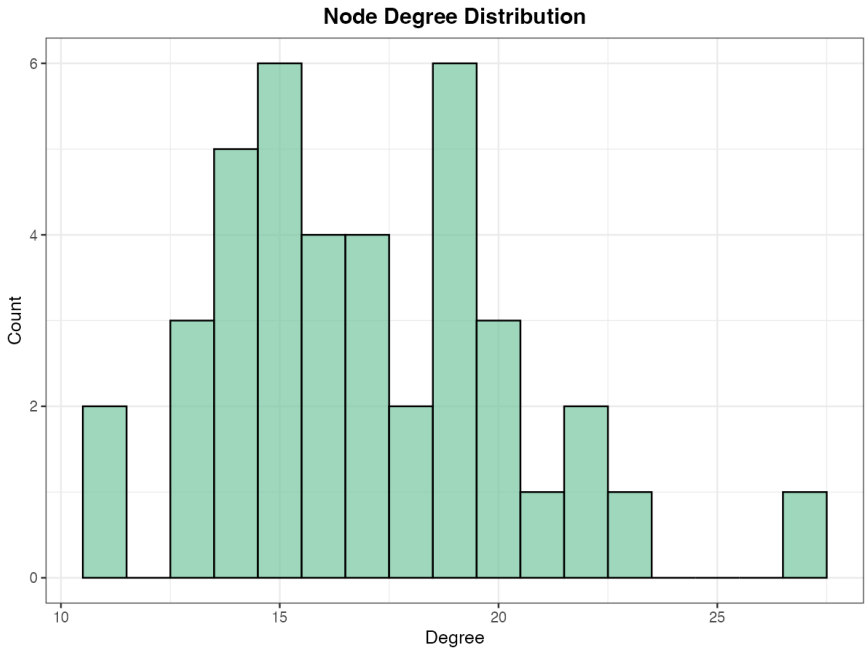
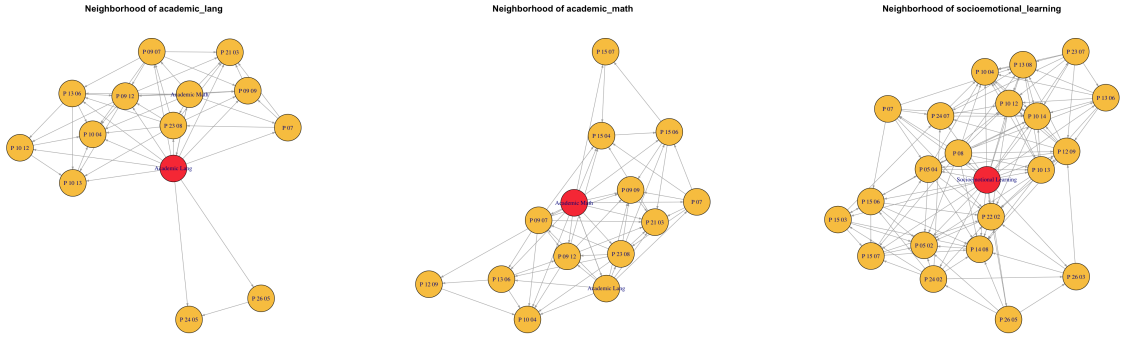


Figure 2: Degree distribution for the questions network (m=4).



(a) `academic_lang` (b) `academic_math` (c) `socioemotional_learning`

Figure 3: Ego neighborhoods for the questions network (m=4).

Questions (m=4).

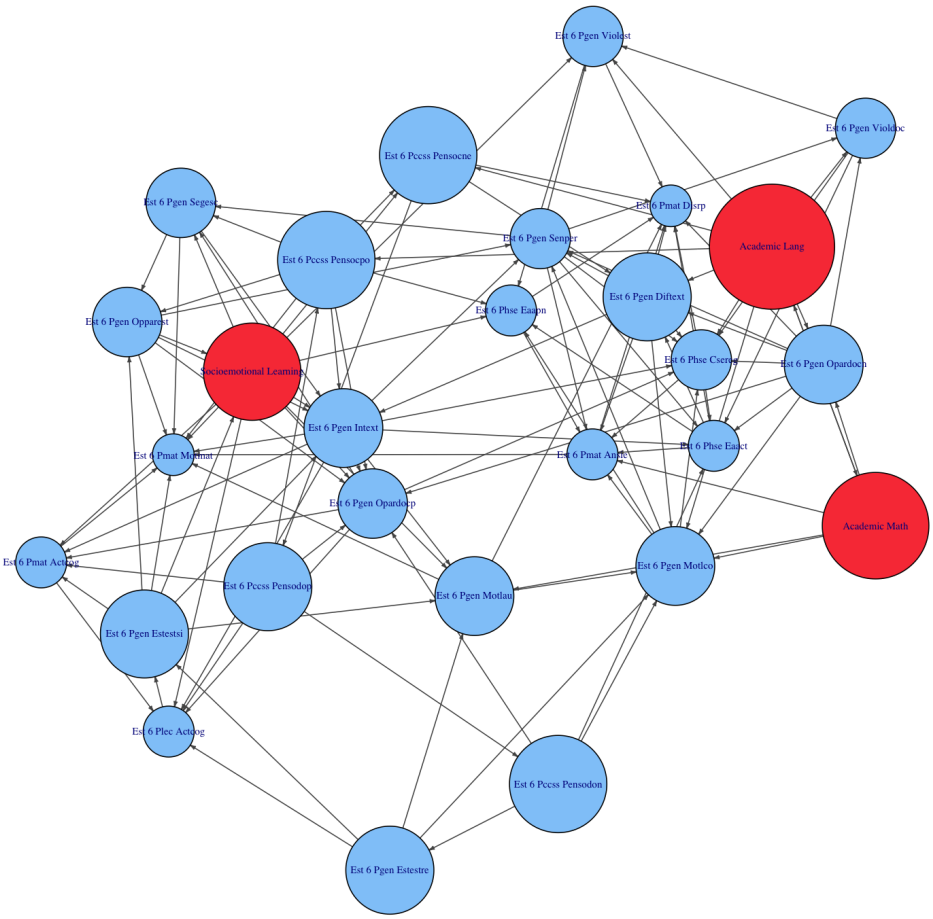


Figure 4: Latent network (m=4): full graph (student questionnaire).

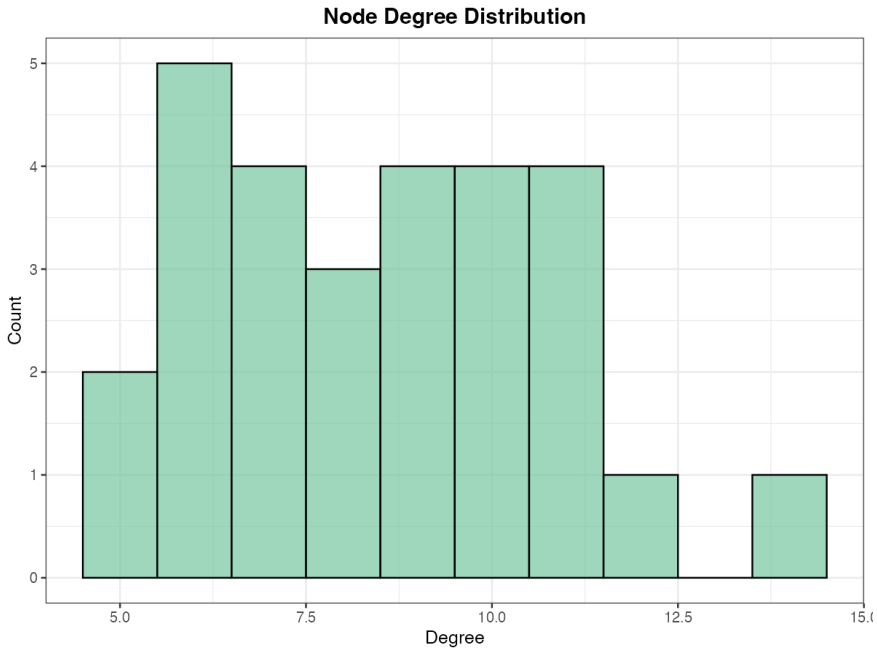
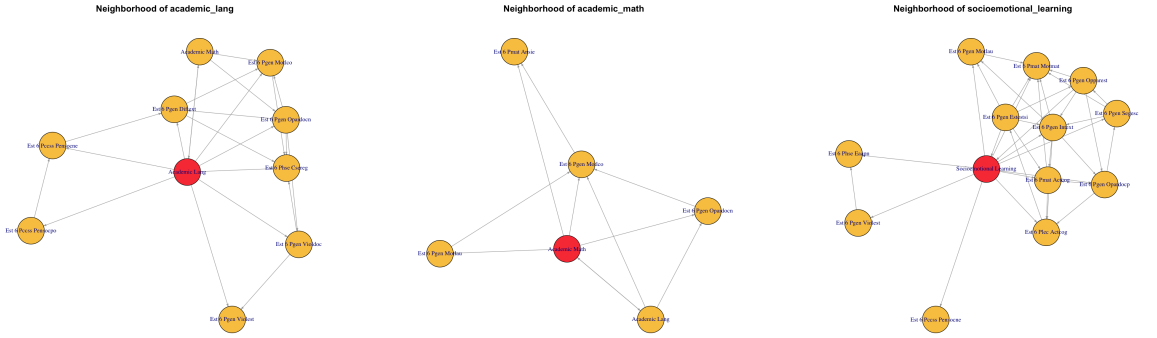


Figure 5: Degree distribution for the latent network (m=4).



(a) `academic_lang` (b) `academic_math` (c) `socioemotional_learning`

Figure 6: Ego neighborhoods for the latent network (m=4).

Latent (m=4).

6 Empirical Driver Lists (Directed Parents)

We extract directed parents for `academic_lang`, `academic_math`, and `socioemotional_learning` by reading the exported edge lists and filtering for incoming arrows. Table 2 summarizes the driver sets across networks and depths.

Table 2: Directed parents (drivers) of key outcomes across networks

Network (depth)	Target	Parents
Questions (m=3)	academic_lang, academic_math, socioemotional_learning	(none detected)
Latent (m=3)	socioemotional_learning	est6pgen_opparest
Questions (m=4)	academic_lang, academic_math, socioemotional_learning	(none detected)
Latent (m=4)	socioemotional_learning	est6pgen_estestsi, est6pgen

7 Interpretation and Recommendations

With depth-1 conditioning, the PC algorithm favors speed and conservatism, leaving some edges undirected where the data do not support a unique orientation. Increasing PC_MAX_COND_SET would likely resolve additional directions at the cost of runtime. In the baseline combined analysis, academic scores appear as upstream drivers of socioemotional_learning; in the questionnaire-specific split, the socioemotional outcome becomes a central emitter among items, indicating the measurement scale influences graph orientation under shallow conditioning. The consistent direction from academic_lang toward academic_math across variants underscores a cross-domain dependency worth further study. Among latent constructs, high-degree hubs such as EST6PGEN_OPPAREST act as integrators of multiple competencies and may present viable levers for targeted interventions.

8 Reproducibility

To regenerate these results:

1. Ensure pipeline/05_pc_algorithm.R is up to date with questionnaire support.
2. Run the script with desired environment variables, for example:

   ```
   PC_QUESTIONNAIRE=estudiante PC_MAX_COND_SET=4 Rscript pipeline/05_pc_algorithm.R
   ```

3. Execute the manual network workflow (for questions and latent) as illustrated within the script to produce edges, node statistics, and figures.