Different Perspectives for Quantifying Curriculum Complexity in Engineering Education Using a Newly Developed R Package

Dr. David Reeping, University of Cincinnati

Dr. David Reeping is an Assistant Professor in the Department of Engineering and Computing Education at the University of Cincinnati. He earned his Ph.D. in Engineering Education from Virginia Tech and was a National Science Foundation Graduate Research Fellow. He received his B.S. in Engineering Education with a Mathematics minor from Ohio Northern University. His main research interests include transfer student information asymmetries, threshold concepts, curricular complexity, and advancing quantitative and fully integrated mixed methods.

Olivia Ryan, Virginia Polytechnic Institute and State University

Olivia Ryan is a Ph.D. student in Engineering Education at Virginia Tech. She holds a B.S. in engineering with a specialization in electrical engineering from Roger Williams University and an M.S. in engineering mechanics from Virginia Tech. Her research interests include understanding curriculum barriers related to mathematics in engineering.

Different Perspectives for Quantifying Curriculum Complexity in Engineering Education Using a Newly Developed R Package

Abstract

This empirical full paper explores new ways to analyze and represent engineering curricula through the concept of Curricular Analytics. The engineering curriculum is notorious for its hierarchical structure of required technical content, the many prerequisites and corequisites requirements for upper-level courses, and numerous accreditation requirements, which make the curriculum complex. Understanding this complexity is essential for improving student progression and success, identifying structural barriers, and supporting curricular redesign. Curricular Analytics has emerged as a valuable tool in engineering education research for understanding structural barriers in degree programs. This analytical framework provides a way to quantify curricula by representing corequisite and prerequisite chains as a network, enabling the use of existing and adapted techniques from fields like social network analysis and graph theory to generate insights into students' course-taking paths and bottlenecks within them.

Studies in the field that use Curricular Analytics often focus primarily on structural complexity, captured through network analysis, to quantitatively assess the importance of individual courses within the program. Recent work has expanded the analytical capabilities by developing a package of functions in the statistical programming platform R that enables examination of curricula from several angles.

To demonstrate the functionality of this package, we draw from two large curricular datasets (n = 494 and n = 155) representing five engineering disciplines (i.e., mechanical, electrical, civil, industrial, and chemical) from 13 institutions and three engineering disciplines (i.e., mechanical, electrical, and civil) from 53 institutions, respectively. We will showcase alternative metrics, including curriculum rigidity (a simpler characterization of structural complexity), core collapse sequence (a visual approach to understanding the curricular design patterns in a curriculum), bottleneck courses (an alternative view of cruciality), and deferment factor (which quantifies how failing a course influences the students' time to degree).

This package offers new ways to analyze engineering curricula, and this work provides descriptive representations of alternative curriculum complexity measures and demonstrates their usefulness across two large datasets of engineering programs. Leveraging these metrics can be key to future research and practice focused on engineering curriculum design.

1. Introduction and Background Information

The engineering curriculum requires students to acquire significant technical knowledge through closely connected courses. As a result, the engineering curriculum is known for its hierarchical structure, with many required courses and numerous prerequisite and corequisite requirements that must also align with accreditation requirements. These requirements lead to a tightly interconnected curriculum that can be challenging to navigate. Such complexity can pose significant challenges for engineering students, especially given the central role of math placement in engineering [1], [2], [3], the challenges of transferring into engineering programs [4], and the low persistence and graduation rates in engineering [5].

Understanding curriculum complexity is essential for improving student progression and success, identifying structural barriers, and supporting curricular redesign [6], [7]. Curricular Analytics has emerged as a valuable tool in engineering education research for understanding structural barriers in degree programs. Curricular Analytics is a broad approach to studying curricula quantitatively by representing corequisite and prerequisite chains as a network [8]. This framework enables the use of existing and adapted techniques from fields such as social network analysis and graph theory to generate insights into students' course-taking paths and bottlenecks within them.

Curriculum Analytics is a growing area of research in engineering education because it can provide insights into student outcomes, curriculum changes, and disciplinary patterns. For example, research surrounding curriculum complexity shows that when curriculum changes are made to sensitive course patterns, graduation rates increase [7]. Further, a significant amount of work in engineering education research has used Curricular Analytics to study issues particularly salient for transfer students [9], [10], post-graduation outcomes [11], as well as differences across majors [11], [12], [13], and across institutions [14], [15], [16], [17].

Currently, a small set of metrics is used to characterize curriculum complexity; however, other metrics have been proposed that offer alternative insights into the curriculum and can help researchers address different questions, such as issues related to the timing of when courses are offered (i.e., [18], [19]). As such, this paper seeks to add to the literature by highlighting the functionality and value of the other curriculum complexity metrics, as identified in a previous scoping review [20]. These metrics can provide insight into existing degree patterns and are used to inform future research or curriculum reform. Thus, the purpose of this paper is to demonstrate the functionality of a newly developed R package for examining curricula in multiple ways, beyond structural complexity, by drawing on two large curricular datasets and highlighting the insights these metrics provide for understanding engineering degree programs. The following research questions guide our paper:

> **RQ:** What alternative curriculum complexity metrics can be used to examine engineering curricula beyond structural complexity? How do these alternative metrics contribute to our understanding of engineering degree programs?

Although this paper is exploratory in nature, we expect that evaluating these metrics will provide insights into differences across engineering curricula and offer readers a more holistic understanding of the complexities of the engineering curriculum.

1.1 Development of R Package

To facilitate the analyses in this paper, an R package was developed and released through CRAN for broader use [21]. The package is built using base R functionality to limit dependencies; the only required package is *igraph* for network analysis [22]. The package includes functionality to perform standard Curricular Analytics calculations (e.g., blocking factor, delay factor, cruciality) and alternative metrics proposed in the literature (e.g., deferment factor, bottlenecks), based on a scoping review of papers that build upon or use Curricular Analytics calculations.

2. Methods

To demonstrate the functionality of the R package, we draw on two curricular datasets developed as part of larger projects. In the following sections, we explain each data source and how the data were analyzed.

2.1 Data Sources

The first dataset was developed in the context of an NSF-funded broadening participation in engineering project to connect the Multiple-Institution Database for Investigating Engineering Longitudinal Development (MIDFIELD) to relevant curricular data, enabling the study of relationships between student outcomes and curricular structures [23]. The majors represented in the dataset are Mechanical, Civil, Chemical, Electrical, and Industrial, which come from 13 institutions in MIDFIELD. These 13 institutions were chosen based on their most recent entry in the dataset being at least in the 2010s. The dataset is longitudinal; each program contains 10 years' worth of plan-of-study information to examine how complexity changes over time. Though not all institutions offered all five disciplines, or did so only for part of the time period examined, the total sample size is 494 plans of study representing 53 distinct programs. The full scope of data collection is described elsewhere [23], [24].

The second dataset comes from a dissertation in engineering education that quantifies and compares curriculum complexity and the role of math in engineering across multiple disciplines and institutions. The dataset includes curriculum plans for three engineering disciplines (i.e., mechanical, electrical, and civil) across 53 institutions (n = 155). The institutions represent 48 of the 50 institutions that awarded the most engineering bachelor's degrees in 2023, based on ASEE *By The Numbers* [25], plus five additional smaller institutions used as counter cases. The curriculum plans in the dataset include information from the 2023, 2024, or 2025 catalog year, depending on which plan was most recently available.

To showcase these curriculum metrics and help us answer our research question, we disaggregate the two datasets across disciplines and provide summary statistics for each metric. We also provide one test case for each metric to illustrate. Moreover, because the NSF MIDFIELD dataset has many overlapping institutions and disciplines due to its longitudinal nature, we used only the most recent catalog year to avoid any single school from skewing the findings.

2.2 Data Analysis

Using the new R package, we chose to highlight five metrics: structural complexity, curriculum rigidity, bottlenecks, deferment factor, and core collapse, to provide more insights into curriculum metrics.

Structural complexity is perhaps the most widely used metric for examining curriculum complexity because it summarizes a program's characteristics in a single number. Structural complexity is the sum of each course's cruciality in the degree plan. A course's cruciality is made up of the blocking factor (i.e., how many courses are blocked for students to take if that course is failed) and the delay factor (i.e., the length of the longest prerequisite chain to which the course belongs). A higher score indicates a more complex curriculum, while a lower score indicates a less complex one. Although this paper seeks to showcase metrics beyond structural complexity, we chose to include structural complexity as a reference for the other metrics.

Another summary score capturing a program's complexity is *curriculum rigidity* (C_r), which is a much simpler calculation. This metric borrows the beta index from graph theory, which is the ratio of edges to vertices. In this case, the edges represent prerequisites and corequisites, and the vertices represent the courses. A curriculum rigidity value greater than 1 indicates a more complex curriculum, and a value below 1 indicates a less complex one. Equation (1) demonstrates how to calculate curriculum rigidity, where $|V|$ is the number of courses in the curriculum and $|E|$ is the number of prerequisite/corequisite relationships between the courses.

$$C_r = \frac{|E|}{|V|} \tag{1}$$

Some courses serve as significant barriers to student progression, often referred to as weed-out or gatekeeper courses. We can determine which courses act as barriers by using the *bottleneck* (B) metric. The bottleneck metric is calculated using thresholds related to prerequisites, post-requisites (the number of courses the course serves as a prerequisite for), and total connections, which can be adjusted to reflect how the user defines a bottleneck. In our case, we define a bottleneck course as one with at least 3 prerequisites, 3 postrequisites, and 5 total connections, based on the suggested values from Wigdhal et al. [26]. Expressed formally, equation (2) shows how we would determine whether a course (v_{ij}) is a bottleneck using the logical operator OR ($\vee$) and inequalities, where *a* represents the minimum number of prerequisites, *b* represents the minimum number of postrequisites, and *c* represents the total number of connections.

$$B(v_{ij}: a, b, c) = \left[\deg^-(v_{ij}) \geq a\right] \vee [\deg^+(v_{ij}) \geq b)] \;\vee \left[\deg(v_{ij}) \geq c\,\right] \tag{2}$$

Failing a course can become a significant barrier to student progression, especially for timely graduation. Therefore, the *deferment factor (D_c)* considers term information, unlike other measures. Equation (3) presents the formula for calculating the deferment factor for an individual course, where *k* represents the maximum number of allowable failures before exceeding the expected program completion time. If a course cannot be failed without extending a student's time to degree, then the deferment factor will be 1. A deferment factor less than 1 indicates a more flexible course.

$$D_c(v_{ij}) = \frac{1}{k+1} \tag{3}$$

Finally, the *core collapse sequence* is a method for exploring curricular design patterns embedded within a program. It tracks how the network changes as courses are progressively removed based on their connectivity (i.e., the number of connections). At each step, we remove courses that meet a given connectivity threshold, starting with those with 0 connections and incrementally increasing the threshold until all courses are removed. Each value in the core collapse sequence represents the proportion of courses removed at that step. So, the full sequence sums to 1. A smooth, gradual decline in this sequence suggests a more uniform network structure, where courses have relatively even connectivity. In contrast, sharp drops or plateaus indicate uneven structure, often reflecting dense clusters of prerequisite relationships or tightly coupled substructures. We can also visualize the core collapse process at each iteration, which provides additional insight into the program's key structural features.

3. Using the R Package

To illustrate the functionality of the R package and highlight the alternative curricular complexity metrics, we will highlight one example for each metric and summary statistics across the different disciplines represented in the study. The example we highlight is Mechanical Engineering at the University of Maryland, College Park, which will hereafter be referred to as “the test case.” For readers interested in replicating the results for their own programs, we include a portion of an example curriculum plan and code for each metric in Appendix A (Figure 2).

3.1 Structural Complexity

Structural complexity is the most commonly used metric for understanding curriculum complexity. We share this metric as a baseline for understanding and conceptualizing the other curriculum metrics. For the test case, the plan's structural complexity is 369. Figure 1 below shows a plot of the example plan of study. The black solid lines represent the prerequisite requirements, and the gray dotted lines represent the corequisite requirements.

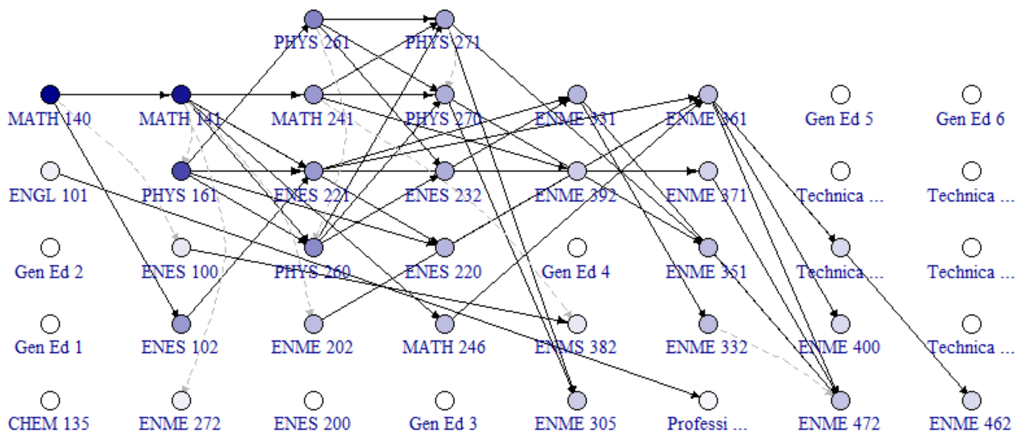

Figure 1: Plot of Plan of Study for the Test Case

A summary of structural complexity values across disciplines is presented in Table 1 below. It can be shown that chemical engineering (*M* = 451.6, *SD* = 162.0) has the highest average structural complexity score, and industrial engineering (*M* = 238.7, *SD* = 42.4) has the lowest; however, we have significantly fewer data points for those disciplines. Across civil, mechanical, and electrical engineering, mechanical engineering has the highest average structural complexity score (M = 388.6, SD = 119.9), and civil engineering has the lowest (*M* = 288.8, *SD* = 100.8).

Table 1: Descriptive Statistics of Structural Complexity Disaggregated by Discipline

Discipline	*n*	Mean	SD	Min	Median	Max
Chemical	7	451.6	162.0	278	429	739
Civil	61	288.8	100.8	127	271	526
Electrical	66	346.2	125.9	129	341.2	906.7
Industrial	7	238.7	42.2	197	221	296
Mechanical	65	388.6	119.9	178	370	743

3.2 Curriculum Rigidity

Curriculum rigidity is a simpler metric than structural complexity, which quantifies the curriculum's complexity with fewer inputs. In this case, unlike structural complexity, where an ideal range for its value is less clear, a value greater than 1 indicates a more complex network, meaning the total number of requisites exceeds the total number of courses. For the test case, the curriculum rigidity score is 1.2, indicating a somewhat more complex network than average.

A summary of curriculum rigidity values across disciplines is presented in Table 2 below. Industrial ($M = 0.88$) and civil ($M = 0.96$) engineering have curriculum rigidity values below 1, indicating they are less complex. Meanwhile, chemical ($M = 1.16$), electrical ($M = 1.05$), and mechanical ($M = 1.26$) have average values above 1, with mechanical being the highest, indicating more complex degree plans. Curriculum rigidity follows similar trends to structural complexity, although chemical engineering had the highest value for structural complexity. Moreover, we observe the highest rigidity in an electrical engineering program, at more than twice the baseline of 1, whereas the lowest rigidity was seen in civil engineering.

Table 2: Descriptive Statistics of Curriculum Rigidity Disaggregated by Discipline

Discipline	*n*	Mean	SD	Min	Median	Max
Chemical	7	1.16	0.17	0.93	1.16	1.39
Civil	61	0.96	0.28	0.41	0.93	1.71
Electrical	66	1.05	0.27	0.52	1.00	2.05
Industrial	7	0.88	0.16	0.60	0.91	1.07
Mechanical	65	1.26	0.25	0.47	1.26	1.71

3.3 Core Collapse

Next, the core collapse metric helps us understand what design patterns are embedded within a program. Core collapse builds on the idea of k-cores in graph theory. A k-core is a subnetwork in which every vertex has at least k connections to other vertices. In this context, core collapse is the procedure of removing courses with more and more connections until every course is eliminated. For each iteration of removing courses, the proportion of courses removed is recorded. For our test case, the sequence of proportions of courses removed is: 0.275, 0.125, 0.175, 0.250, and 0.175. An illustration of the plot of the plan of study of each iteration is shown in Figure 3 in Appendix B. This metric provides another way to visualize the curriculum's structure and the key courses and sequences on which the curriculum plan relies.

Since a diagram cannot be shown for every curriculum plan in our dataset, we present summary statistics for the core collapses. In Table 3, we have the average length of the core collapse sequence, showing how many iterations, on average, curriculum plans for each discipline take to collapse completely. It can be shown that, across disciplines, degree plans typically take 5 to 6 iterations until there are no vertices left.

Table 3: Descriptive Statistics of Number of Iterations in Core Collapse Disaggregated by Discipline

Discipline	*n*	Mean	SD	Min	Median	Max
Chemical	7	5.71	0.49	5	6	6
Civil	61	4.95	0.76	4	5	7
Electrical	66	5.12	0.64	4	5	7
Industrial	7	4.71	0.49	4	5	5
Mechanical	65	5.60	0.70	4	6	7

3.4 Bottlenecks

The bottleneck function provides an opportunity to identify courses that tend to be significant barriers to student success in a more direct fashion compared to cruciality. With this metric, we can identify bottleneck courses based on a predetermined minimum number of prerequisites, the minimum number of courses the course serves as a prerequisite for (postreqs), and the minimum number of connections. In our analysis, we use three prerequisites, three postreqs, and five connections as parameters, based on the literature, which provides these as suggested values [26]. For the test case, 12 courses were identified as bottlenecks based on the predetermined characteristics. Many of the bottleneck courses are unsurprising, such as Calculus and Physics, which are often key to advancing the curriculum. Additionally, some upper-level required classes emerged as bottleneck courses, such as Fluids and Transportation (see Table 4). These bottleneck courses can be spotted in the output of the core collapse sequence. Many of the bottleneck courses are those left in later iterations of the core collapse plots, allowing researchers and practitioners to unravel the interconnections among requirements across programs.

Table 4: List of Bottleneck Courses for the Test Case

Course Abbreviation	Course Title
MATH 140	Calculus 1
PHYS 161	General Physics: Particle Dynamics
MATH 141	Calculus 2
PHYS 260	General Physics: Electricity, Magnetism, and Thermodynamics
PHYS 261	General Physics: Mechanics, Vibrations, Waves, Heat (Lab)
ENES 220	Mechanics 2
ENCE 202	Engineering Drawings and Design for Civil and Environmental Engineers
ENCE 305	Fundamentals of Engineering Fluids
ENCE 303	Probability and Statistics for Civil and Environmental Engineers
ENCE 336	Environment and Water 1: Introduction to Environmental Engineering
ENCE 383	Transportation Systems 1
ENCE 464	Civil and Environmental Engineering Design 1

To show differences across disciplines, we provide a summary of the average number of bottleneck courses from each degree plan in Table 5 below. The statistics show that civil and industrial engineering have fewer bottleneck courses than chemical and mechanical engineering. This finding aligns with our findings on structural complexity and curriculum rigidity, showing that more complex degree plans tend to have more bottleneck courses.

Table 5: Descriptive Statistics of Number of Bottleneck Courses Disaggregated by Discipline

Discipline	*n*	Mean	SD	Min	Median	Max
Chemical	7	15.10	4.52	11	16	23
Civil	61	8.10	4.97	1	7	24
Electrical	66	9.40	4.56	2	8.5	29
Industrial	7	7.40	2.64	4	8	11
Mechanical	65	13.60	4.50	4	14	22

3.5 Deferment Factor

Finally, the deferment factor provides insight into how the specific courses can affect student progress. The deferment factor considers each course and the expected number of terms to complete the program as designed. If a course cannot be failed without extending a student's time to degree, then the deferment factor will be 1. For the test case, we show the deferment factor for each bottleneck course. Since these courses are highly embedded in the curricular structure, they are more likely to extend students' time to degree if they are failed. It can be shown that many of the bottleneck courses have a deferment factor of 0.5, implying the course cannot be failed more than once without extending the student's time to degree (see Table 6). Adding the deferment factor provides another lens through which to approach the bottleneck courses by using timing in the plan of study. Although there is overlap, with many bottleneck courses having a deferment factor of 0.5, that doesn't mean they are synonymous metrics. A course with a deferment factor of 1 means that failing the course will extend students' time-to-degree, whereas a bottleneck course is just a heavily interconnected course.

Table 6: Deferment Factors of Bottleneck Courses for the Test Case

Course Abbreviation	Course Title	Deferment Factor
MATH 140	Calculus 1	0.50
PHYS 161	General Physics: Particle Dynamics	0.50
MATH 141	Calculus 2	0.50
PHYS 260	General Physics: Electricity, Magnetism, and Thermodynamics	0.33
PHYS 261	General Physics: Mechanics, Vibrations, Waves, Heat (Lab)	0.33
ENES 220	Mechanics 2	0.50
ENCE 202	Engineering Drawings and Design for Civil and Environmental Engineers	0.17
ENCE 305	Fundamentals of Engineering Fluids	0.25
ENCE 303	Probability and Statistics for Civil and Environmental Engineers	0.50
ENCE 336	Environment and Water 1: Introduction to Environmental Engineering	0.50
ENCE 383	Transportation Systems 1	0.50
ENCE 464	Civil and Environmental Engineering Design 1	0.50

To understand how the deferment data looks across the dataset, we examined the average count of the maximum value (i.e., 1). Then, for each curriculum plan, we determined the percentage of courses with a deferment factor of 1. In Table 7, it can be shown that industrial engineering (M = 0.17) had the lowest average percentage of deferment factor values equal to 1. Meanwhile, Chemical (M = 0.43) had the highest average percentage of deferment factor values equal to 1.

Table 7: Descriptive Statistics of Percent of Deferment Factor Values Equal to 1 Disaggregated by Discipline

Discipline	*n*	Mean	SD	Min	Median	Max
Chemical	7	0.43	0.04	0.37	0.43	0.49
Civil	61	0.22	0.10	0.02	0.19	0.50
Electrical	66	0.27	0.11	0.05	0.28	0.50
Industrial	7	0.17	0.12	0.05	0.13	0.42
Mechanical	65	0.32	0.13	0.06	0.32	0.61

In addition to the average deferment factor per plan being 1, we also examined the deferment factor mean. For each plan, we took the average deferment factor and then computed the mean for each discipline. We found that the average deferment factor for all courses in a plan of study hovers between 0.41 and 0.60 (see Table 8). Across the deferment factor findings, we show that, on average, engineering curriculum plans include many courses that can pose challenges to student success and progression.

Table 8: Descriptive Statistics of Deferment Factor Disaggregated by Discipline

Discipline	*n*	Mean	SD	Min	Median	Max
Chemical	7	0.60	0.03	0.58	0.58	0.65
Civil	61	0.44	0.08	0.28	0.43	0.66
Electrical	66	0.49	0.09	0.28	0.50	0.65
Industrial	7	0.41	0.09	0.33	0.40	0.56
Mechanical	65	0.53	0.10	0.31	0.53	0.73

4. Discussion and Limitations

Throughout the showcase of the metrics, readers might notice that many of them are correlated with one another. For example, structural complexity and curricular rigidity are closely related, which is not surprising. Many of the metrics are based on the same network properties, specifically connections between vertices. As such, each metric presents a different lens through which the researcher or practitioner can understand different structural facets of the curriculum they are examining [20].

The suite of structural metrics - including structural complexity, rigidity, cruciality, bottleneck, and deferment factor - provides researchers and practitioners with analytic leverage to interrogate curricular designs from multiple vantage points. These metrics can be used to benchmark degree programs (e.g., via structural complexity and curricular rigidity), identify gatekeeper and bottleneck courses (through cruciality and bottleneck counts), and evaluate the temporal architecture of course scheduling using inflexibility and deferment factors alongside term-weighted cruciality. They further enable targeted analyses around transfer student progress by examining how structural inflexibility and transfer delay factors influence degree momentum, and can support efforts to plan more balanced credit loads across semesters. Collectively, these measures create a structural lens on program design that complements instructional indicators—such as pass rates, grade anomalies, conformity scores, and pass-through effects—and ultimately allow institutions to simulate student flow, assess completion rates, and demonstrate continuous improvement in curriculum planning and outcomes.

A key limitation of this work is the relatively small number of degree plans available for certain disciplines, particularly industrial and chemical engineering, which each included only seven curricula. While the datasets were sufficient for demonstrating the functionality of the R package and generating initial comparative insights, this limited sample size constrains the generalizability of findings within these fields. With so few programs represented, observed patterns may be more sensitive to institution-specific curricular designs than to broader disciplinary norms. As a result, comparisons across these disciplines should be interpreted cautiously, particularly when contrasted with fields represented by a larger number of degree plans. Future work would benefit from expanding the dataset to include a wider range of institutions and program variations, enabling more robust estimation of within-discipline variability and strengthening cross-disciplinary comparisons.

Moreover, our choice of metrics here was limited by the types of data in our two datasets. While the literature offers several transfer-specific metrics (e.g., transfer delay factor, credit loss, and inflexibility factor) that are offered in the package [19], [20], [27], these measures were not relevant to the dataset used in this study because the pathways were exclusively first-time-in-college. Future work can showcase these metrics using a dataset focused on examining the complexity of transfer pathways.

5. Conclusion

This study introduced and illustrated a newly developed R package for alternative curriculum complexity metrics using two large datasets. Across multiple engineering majors, we demonstrated how these metrics can be applied within Curricular Analytics and provided baseline comparisons to support interpretation across programs. The results suggest variation in structural complexity across disciplines, with industrial and civil engineering tending toward lower values than chemical, mechanical, and electrical engineering, indicating that curriculum design patterns differ systematically across fields.

The applied example also illustrates how the package can be used to surface curricula's structural features. In particular, courses that function as potential bottlenecks emerged across both foundational sequences (e.g., Calculus and Physics) and upper-division technical requirements, highlighting how constraints may be distributed throughout degree pathways rather than concentrated in a single stage. Taken together, these findings reinforce the value of examining curricula as structured systems that shape student progression. By providing explicit implementations of alternative complexity measures across two large datasets, this work supports more nuanced comparisons of engineering programs. It offers a foundation for future research on how curricular structure relates to student pathways and program design.

6. References

[1] J. Main and A. Griffith, “The Impact of Math and Science Remedial Education on Engineering Major Choice, Degree Attainment, and Time to Degree,” in *2022 ASEE Annual Conference & Exposition*, 2022.

[2] J. Van Dyken, L. Benson, and P. Gerard, “Persistence in Engineering: Does Initial Mathematics Course Matter?,” presented at the 2015 ASEE Annual Conference & Exposition, Jun. 2015, p. 26.1225.1-26.1225.9. Accessed: Dec. 08, 2022. [Online]. Available: https://peer.asee.org/persistence-in-engineering-does-initial-mathematics-course-matter

[3] O. Ryan, S. Sajadi, S. Barrera, and R. T. Jaghargh, “Understanding the Effects of a Math Placement Exam on Calculus 1 Enrollment and Engineering Persistence,” *Educ. Sci.*, vol. 15, no. 2, Art. no. 2, Feb. 2025, doi: 10.3390/educsci15020154.

[4] A. J. Richardson and D. B. Knight, “Exploring Credit Loss for Engineering Transfer Students,” *Inq. J. Va. Community Coll.*, vol. 28, no. 1, 2025, Accessed: Jan. 20, 2026. [Online]. Available: https://eric.ed.gov/?id=EJ1468116

[5] American Society for Engineering Education, “Engineering by the Numbers: ASEE Retention and Time-to-Graduation Benchmarks for Undergraduate Engineering Schools, Departments and Programs.”

[6] D. Reeping, D. Grote, L. D. McNair, and T. Martin, “Curricular Complexity as a Metric to Forecast Issues with Transferring into a Redesigned Engineering Curriculum,” presented at the 2020 ASEE Virtual Annual Conference Content Access, Jun. 2020. Accessed: Nov. 24, 2023. [Online]. Available: https://peer.asee.org/curricular-complexity-as-a-metric-to-forecast-issues-with-transferring-into-a-redesigned-engineering-curriculum

[7] G. L. Heileman, M. Hickman, A. Slim, and C. T. Abdallah, “Characterizing the Complexity of Curricular Patterns in Engineering Programs,” presented at the 2017 ASEE Annual Conference & Exposition, Jun. 2017. Accessed: Nov. 24, 2023. [Online]. Available: https://peer.asee.org/characterizing-the-complexity-of-curricular-patterns-in-engineering-programs

[8] G. L. Heileman, C. T. Abdallah, A. Slim, and M. Hickman, “Curricular Analytics: A Framework for Quantifying the Impact of Curricular Reforms and Pedagogical Innovations,” Nov. 14, 2018, *arXiv*: arXiv:1811.09676. doi: 10.48550/arXiv.1811.09676.

[9] D. Reeping, D. M. Grote, and D. B. Knight, “Effects of Large-Scale Programmatic Change on Electrical and Computer Engineering Transfer Student Pathways,” *IEEE Trans. Educ.*, vol. 64, no. 2, pp. 117–123, May 2021, doi: 10.1109/TE.2020.3015090.

[10] D. M. Grote, D. B. Knight, W. C. Lee, and B. A. Watford, “Navigating the Curricular Maze: Examining the Complexities of Articulated Pathways for Transfer Students in Engineering,” *Community Coll. J. Res. Pract.*, vol. 45, no. 11, pp. 779–801, Nov. 2021, doi: 10.1080/10668926.2020.1798303.

[11] M. A. Alrizqi and A. Godwin, “Data-Driven Analysis of Engineering Curricula: A Cross-Disciplinary Study of Complexity in Seven Programs and Its Impact on Student Pathways and Career Outcomes,” presented at the 2025 ASEE Annual Conference & Exposition, Jun. 2025. Accessed: Sep. 02, 2025. [Online]. Available: https://peer.asee.org/data-driven-analysis-of-engineering-curricula-a-cross-disciplinary-study-of-complexity-in-seven-programs-and-its-impact-on-student-pathways-and-career-outcomes

[12] K. Davis, M. Chen, and A. Wolf, "Is Curriculum Complexity Related to Study Abroad Participation? A Cross-Major Comparison at One University," in *2024 ASEE Annual Conference & Exposition Proceedings*, Portland, Oregon: ASEE Conferences, Jun. 2024, p. 47706. doi: 10.18260/1-2--47706.
[13] D. R. Waller, "Organizational Factors and Engineering Student Persistence," Ph.D., Purdue University, United States -- Indiana, 2022. Accessed: Sep. 04, 2025. [Online]. Available: https://www.proquest.com/docview/2838334542/abstract/1462B909E05C4AD7PQ/1
[14] D. Reeping and N. Rashedi, "Work in Progess: A Decade-Spanning Longitudinal Study on the Curricular Complexity of Engineering Programs," in *2023 IEEE Frontiers in Education Conference (FIE)*, Oct. 2023, pp. 1–5. doi: 10.1109/FIE58773.2023.10343227.
[15] G. L. Heileman, W. G. Thompson-Arjona, O. Abar, and H. W. Free, "Does Curricular Complexity Imply Program Quality?," presented at the 2019 ASEE Annual Conference & Exposition, Jun. 2019. Accessed: Nov. 24, 2023. [Online]. Available: https://peer.asee.org/does-curricular-complexity-imply-program-quality
[16] G. L. Heileman, H. W. Free, J. Flynn, C. Mackowiak, J. W. Jaromczyk, and C. T. Abdallah, "Curricular Complexity Versus Quality of Computer Science Programs," Jun. 11, 2020, *arXiv*: arXiv:2006.06761. doi: 10.48550/arXiv.2006.06761.
[17] A. Lionelle, M. Quam, C. Brodley, and C. Gill, "Does Curricular Complexity in Computer Science Influence the Representation of Women CS Graduates?," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, in SIGCSE 2024. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 729–735. doi: 10.1145/3626252.3630835.
[18] R. Molontay, N. Horváth, J. Bergmann, D. Szekrényes, and M. Szabó, "Characterizing Curriculum Prerequisite Networks by a Student Flow Approach," *IEEE Trans. Learn. Technol.*, vol. 13, no. 3, pp. 491–501, Jul. 2020, doi: 10.1109/TLT.2020.2981331.
[19] D. Reeping and D. Grote, "Characterizing the Curricular Complexity Faced by Transfer Students: 2+2, Vertical Transfers, and Curricular Change," in *2022 ASEE Annual Conference & Exposition Proceedings*, Minneapolis, MN: ASEE Conferences, Aug. 2022, p. 41462. doi: 10.18260/1-2--41462.
[20] N. Rashedi, D. Reeping, and S. Wei, "A scoping review of methods used to analyze engineering curricula quantitatively with curricular analytics," *Eng. Educ. Rev.*, vol. 4, Mar. 2026, doi: 10.54844/eer.2025.1110.
[21] D. Reeping, *CurricularComplexity: Toolkit for Analyzing Curricular Complexity*. (Jan. 09, 2026). Accessed: Jan. 20, 2026. [Online]. Available: https://cran.r-project.org/web/packages/CurricularComplexity/index.html
[22] G. Csardi, "The igraph software," *InterJournal Complex Syst*, 2006.
[23] D. Reeping, H. Ebrahiminejad, M. W. Ohland, K. J. Reid, and N. Rashedi, "Analyzing the Curricular Complexity of Engineering Programs Across Disciplines and Time," *IEEE Trans. Educ.*, pp. 1–10, 2026, doi: 10.1109/TE.2026.3658085.
[24] D. Reeping, S. M. Padhye, and N. Rashedi, "A Process for Systematically Collecting Plan of Study Data for Curricular Analytics," presented at the 2023 ASEE Annual Conference & Exposition, Jun. 2023. Accessed: Apr. 10, 2026. [Online]. Available: https://peer.asee.org/a-process-for-systematically-collecting-plan-of-study-data-for-curricular-analytics

[25] ASEE, “Engineering & Engineering Technology by the Numbers, 2023,” 2024. [Online]. Available: https://ira.asee.org/wp-content/uploads/2024/10/Engineering-Engineering-Technology-By-the-Numbers-2023-27-October-2024.pdf
[26] J. Wigdahl, G. L. Heileman, A. Slim, and C. T. Abdallah, “Curricular efficiency: What role does it play in student success?,” in *2014 asee annual conference & exposition*, 2014, pp. 24–344.
[27] D. Reeping and D. Grote, “Rethinking the curricular complexity framework for transfer students,” in *2021 ASEE Virtual Annual Conference Content Access*, 2021.

7. Appendix

7.1 Appendix A

Figure 2: Portion of Example Curriculum Plan

	A	B	C	D	E	F	G	H	I	J
1	Course	Term	Prerequisites	Corequisites	Credits	Name				
2	ENES 100	1		MATH 140	3	Introduction to Engineering Design				
3	CHEM 135	1			3	General Chemistry for Engineers				
4	MATH 140	1			4	Calculus 1				
5	Science Elec	1			3					
6	ENGL 101	1			3	Academic Writing				
7	PHYS 161	2		MATH 141	3	General Physics: Particle Dynamics				
8	MATH 141	2	MATH 140		4	Calculus 2				
9	ENES 102	2	MATH 140		3	Mechanics 1				
10	ENES 200	2			3	Technology and Consequences: Engineering, Ethics, and Humanity				
11	Gen Ed 1	2			3	General Education				
12	PHYS 260	3	PHYS 161, M/	PHYS 261	3	General Physics: Electricity, Magnetism, and Thermodynamics				
13	PHYS 261	3	PHYS 161		1	General Physics: Mechanics, Vibrations, Waves, Heat (Lab)				

Example Code

```
install.packages("CurricularComplexity")

library("readxl")
library("CurricularComplexity")

# Set the working directory and load the curriculum plan into a data frame
setwd("Name of directory where file is saved")
CurriculumName <- read_excel("Name of file")

# Create a plan of study for the curriculum
CurriculumName_pos <- create_plan_of_study(
                    Course = CurriculumName_pos$Course,
                    Term = CurriculumName_pos $Term,
                    Prereq = CurriculumName_pos $Prerequisites,
                    Coreq = CurriculumName_pos $Corequisites
                    )

# plots the plan of study as a network
plot_plan_of_study(CurriculumName_pos)

# calculate the structural complexity metric
structural_complexity(CurriculumName_pos)

# calculate the curriculum rigidity metric
curriculum_rigidity(CurriculumName_pos)

# calculate the curriculum rigidity metric
find_bottlenecks(CurriculumName_pos, min_prereq = 3, min_postreq = 3, min_connections = 5,
include_coreqs = TRUE)

# calculate the deferment factor for a specific course, for example, "MATH 140"
```

```r
deferment_factor(CurriculumName_pos, "MATH 140", 8)

# run the core collapse metric
example_core_collapse <- core_collapse(CurriculumName_pos)

# print out the core collapse sequence
example_core_collapse$sequence

# plot the core collapse for a specific iteration, n
kcores <- example_core_collapse$kcores
plot_plan_of_study(kcores[[n]])
```

7.1 Appendix B

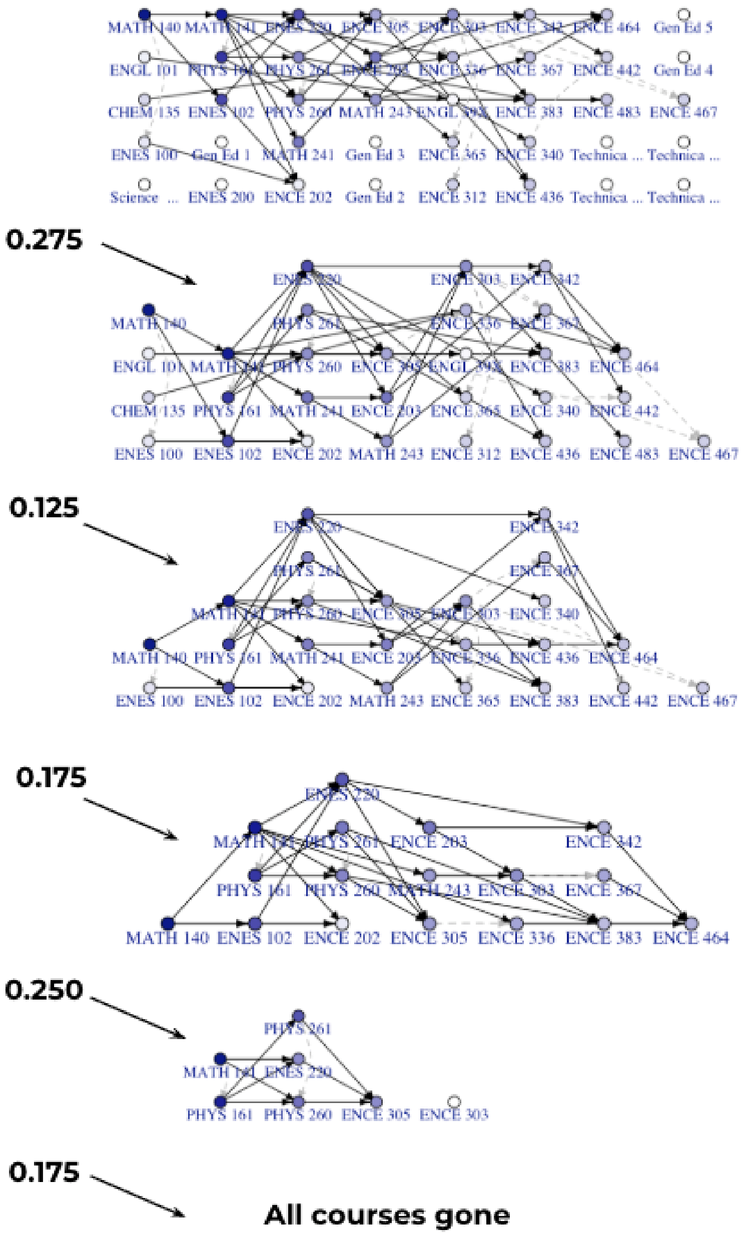

Figure 3: Flow of Core Collapse Sequence