2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering
Education for 130 Years
Paper ID #38560

A Process for Systematically Collecting Plan of Study Data for Curricular Analytics

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.

Mr. Sushant Makarand Padhye, University of Cincinnati

Sushant Padhye is a freshman majoring in electrical engineering and an Undergraduate Research Assistant at the University of Cincinnati.

Nahal Rashedi

A Process for Systematically Collecting Plan of Study Data for Curricular Analytics

Abstract

This theory paper describes challenges and opportunities with analyzing engineering curricula using the *Curricular Analytics* framework. We offer a data collection framework for systematically transforming engineering plans of study into network data at scale. Introduced by Heileman and colleagues in 2017, the Curricular Analytics framework enables researchers and practitioners to quantify the interconnectedness of their prerequisite structures to unveil gatekeeper courses and forecast the impact of curricular policies or changes using network-analytic metrics. These metrics can be calculated using only available data; all one needs to do is transform a plan of study into a list of courses, prerequisites, and corequisites. However, larger projects that examine institutional, disciplinary, and temporal differences will likely face difficulties when wrangling with the details of diverse organizational contexts. This paper outlines the data entry processes developed by drawing from a research group's Microsoft Teams communications for a National Science Foundation-sponsored project to explore decade-long trends in curricular complexity across institutions in the Multi-Institution Database for Engineering Longitudinal Development (MIDFIELD). We anticipate these suggestions will streamline data collection for similar large-scale projects that employ Curricular Analytics as their analytical approach in the future.

Background

Curricular Analytics involves quantifying a curriculum's complexity to correlate its metrics with proxies for student success, often degree completion rates, to understand the relationship between curricular design or policy and student outcomes. To accomplish the quantification, we represent a plan of study outlining the coursework requirements a student must complete in order to earn a degree as a network. In the network, courses are represented as vertices (or nodes), and the prerequisite relationships among them are given by directed edges (arrows). This data type allows us to calculate a suite of metrics drawn from the pool of techniques developed in other fields, like social network analysis, that can help us capture "complexity" in some meaningful way. First appearing in its most recognizable form in work by Wigdahl as the idea of "curricular efficiency" [1], Heileman et al. [2] provide a thorough treatment of the possible quantities that form Curricular Analytics.

Curricular complexity is divided into two components: instructional complexity and structural complexity [2]. Instructional complexity attempts to capture the latent factors of the curriculum, such as course difficulty and instructional quality, but it is currently only proxied by a course's pass rate. Explicit advancements in the idea of instructional complexity are almost non-existent, with the exception of Waller, who reframed course difficulty using the concept of grade anomalies and found it to be a more robust metric than individual course DFW rates in his study of organizational factors' impact on student success [3]. Structural complexity has been explored to a greater extent, likely because of the ability to access public data and construct simulations with relative ease [4]–[8]. Structural complexity is concerned with assigning a score to a curriculum based on the interconnectedness of program requirements (i.e., pre- and corequisites).

Among the metrics proposed for structural complexity, two have become central to how complexity is calculated. These are visualized in Figure 1.

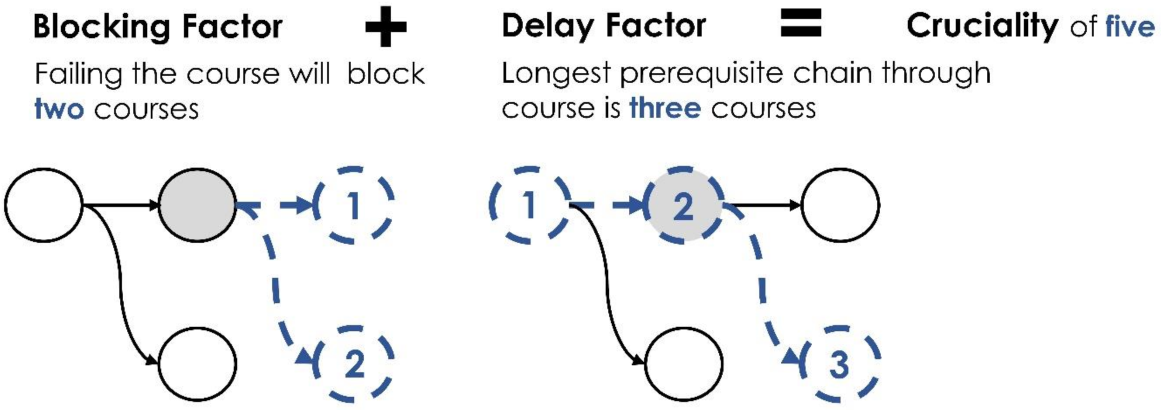

Figure 1. Calculating course cruciality using the blocking and delay factor of the gray course

The first metric is the blocking factor, which is found by counting the number of courses inaccessible to a student if the course is failed. The second metric is the delay factor, the length of the longest prerequisite chain flowing through the course. Adding these two values together yields the cruciality, a local measurement of how entangled a course is in the prerequisite structures of the plan of study and how essential it is to complete. The cruciality can be used to find potential bottlenecks, or gatekeeper courses, in the curriculum [6], [9]. For a global measurement, simply add all the crucialities together – this value is called the *structural complexity* of the curriculum. These global measurements can be used to compare curricula within and across different strata, such as majors within other colleges [4] or engineering disciplines [10]. Other efforts have also emerged to apply to connect curricular complexity with topic-level dependencies between courses [11], incorporate a probabilistic student flow approach [12], and extend the framework to be sensitive to transfer student issues [13], [14].

Although the data requirements are minimal, there are distinct challenges to employing this framework across disciplines and institutions – especially if longitudinal analyses are planned. For example, curricula are not always completely defined, leaving space for students to select electives with varying degrees of flexibility. Without specifying these electives, the curriculum's complexity may be underestimated. Moreover, there are deeper data entry considerations; prerequisites can often be a complicated series of ANDs and ORs, with language like "at least" or "X of the following." These configurations do not lead to obvious network representations. Finally, finding accurate plan of study information, even for recent years, can be challenging. Prerequisite and course information can be inaccurate or may not have been appropriately maintained by the appropriate institutional office. As Curricular Analytics is applied more broadly, reflecting on current practices and considering how this framework can be expanded to capture more nuanced curricular representations is valuable.

Research Aim

This paper recounts the obstacles encountered during the data collection process for a longitudinal multi-institution project employing Curricular Analytics and offers the data conventions we developed to overcome those obstacles. We outline these procedures not only for transparency, but also to assist other researchers and practitioners who want to use the

Curricular Analytics framework at scale. Given the lack of formal guidance on handling plan of study data for broader projects, we contend this work can become a resource to fill the current gap in standard practices for proper data entry to analyze curricula.

Drawing Insights from a Broader Longitudinal Project on Curricular Complexity

This work is derived from a larger project focused on quantifying plans of study for five engineering disciplines (Civil Engineering, Electrical Engineering, Mechanical Engineering, Chemical Engineering, and Industrial Engineering) to compare the complexity of such programs across the United States. The sampling frame, in this case, was the Multi-Institution Database for Engineering Longitudinal Development (MIDFIELD) [15]. Data collection for the larger project was completed in January 2023. In Fall 2022, five undergraduate research assistants in the College of Engineering and Applied Science and a PhD student in engineering education entered data from course catalogs over the course of a decade from the most recent record for thirteen schools in MIDFIELD, resulting in 494 plans of study. To facilitate team communication and data quality checks, we used a channel in Microsoft Teams. The research team was encouraged to talk with one another and ask questions if they ran into issues during data collection.

One of the major outputs of the NSF project is an R package that will allow other researchers to interact with the dataset we created and use more customized functions to explore different dimensions of curricular complexity. R is also open source, making the platform and package more accessible to the community. We chose to write our package in R because of the existing packages for analyzing data from MIDFIELD, namely midfieldr [16] and midfielddata [17]. The midfieldr contains ready-to-go functions for properly processing data at the student level, and the second package is a stratified random sample from MIDFIELD for users to practice on and explore. The data produced from this project will be made available in a similar format. We anticipate the output synergizing with the broader goal of expanding access to and participation in MIDFIELD's development [18].

Data Collection

Although originally intended for project communication alone and standardization to ensure process reliability [19], we found our Microsoft Teams chats to be a valuable source of practical issues that other teams might encounter when conducting a similar project. The data source for this work was the set of chats from our team's communication regarding data entry within the platform, Microsoft Teams. Due to licensing policies with the university-sponsored Microsoft Teams account, it was not feasible to export all the chats in the channel through a request to the university IT department. Instead, after expanding all chats to ensure longer messages and pictures were not cut off, the Chrome plugin GoFullPage created a pdf of the channel using the web-browser version of Teams. One minor inconvenience of this method is that the resulting pdf was not searchable, meaning that all relevant questions needed to be labeled by hand and referenced by number using a spreadsheet. After filtering out non-data entry questions (e.g., questions about submitting timesheets or meeting time updates), we were left with 88 questions.

Analysis

We employed descriptive coding to split the questions into categories where similar or repeated questions could be grouped [20]. To create more generalized questions that removed specific institutional context, questions within each category were processed using the constant comparative method [21] to consolidate similar inquiries into one unique question.

The research assistants, both undergraduate and graduate, who entered the data in Fall 2022 were involved in the synthesis of these generalized questions, and their perspectives shared in weekly meetings were used as peer debriefing [22]. Among these questions, we selected the inquiries related to general data entry and created a flowchart to summarize a process for entering curricular data in similar projects.

A General Process for Collecting and Extracting Plan of Study Data

Through our data collection processes in Fall 2022, including the discussions and questions posed during data entry, we have assembled the following considerations for collecting plans of study for use with the Curricular Analytics framework. These considerations include methodological issues and general process inconsistencies that can emerge. We divided it into three components: pre-processing (Figure 2), processing (Figure 3), and post-processing (Figure 4). In pre-processing, we made a pdf of the source material, such as the website or catalog page, using the naming convention: *InstitutionName_CatalogYear_DisciplineName_POS.pdf*. For the data file, we used the convention, *InstitutionName_CatalogYear_DisciplineName_Base.csv,* for the specified courses and *InstitutionName_CatalogYear_DisciplineName_Elective.csv* for electives. The CatalogYear will be whatever year range is on the title of the catalog. For example, if the catalog year is 2021-2022, then the CatalogYear will be 2122. The pre-processing step considers cases where summer or winter terms appear and if courses are not assigned to terms or semesters (such as by year). The processes in Figure 5 handle data processing, including the conventions we established earlier for nontrivial prerequisite structures, embedded labs, and additional enrolment requirements. In Figure 6, we clean up any loose ends by checking for mutual corequisites and sorting courses without term assignments because courses were not specified by semester by rebalancing courses free to move until credit hour distributions fall between the minimum to be enrolled and below the maximum.

In the next sections, we will elaborate on the data collection and entry questions that we encountered during Fall 2022 by splitting up the discussion into pre-processing, processing, and post-processing – which will map to our flow diagram.

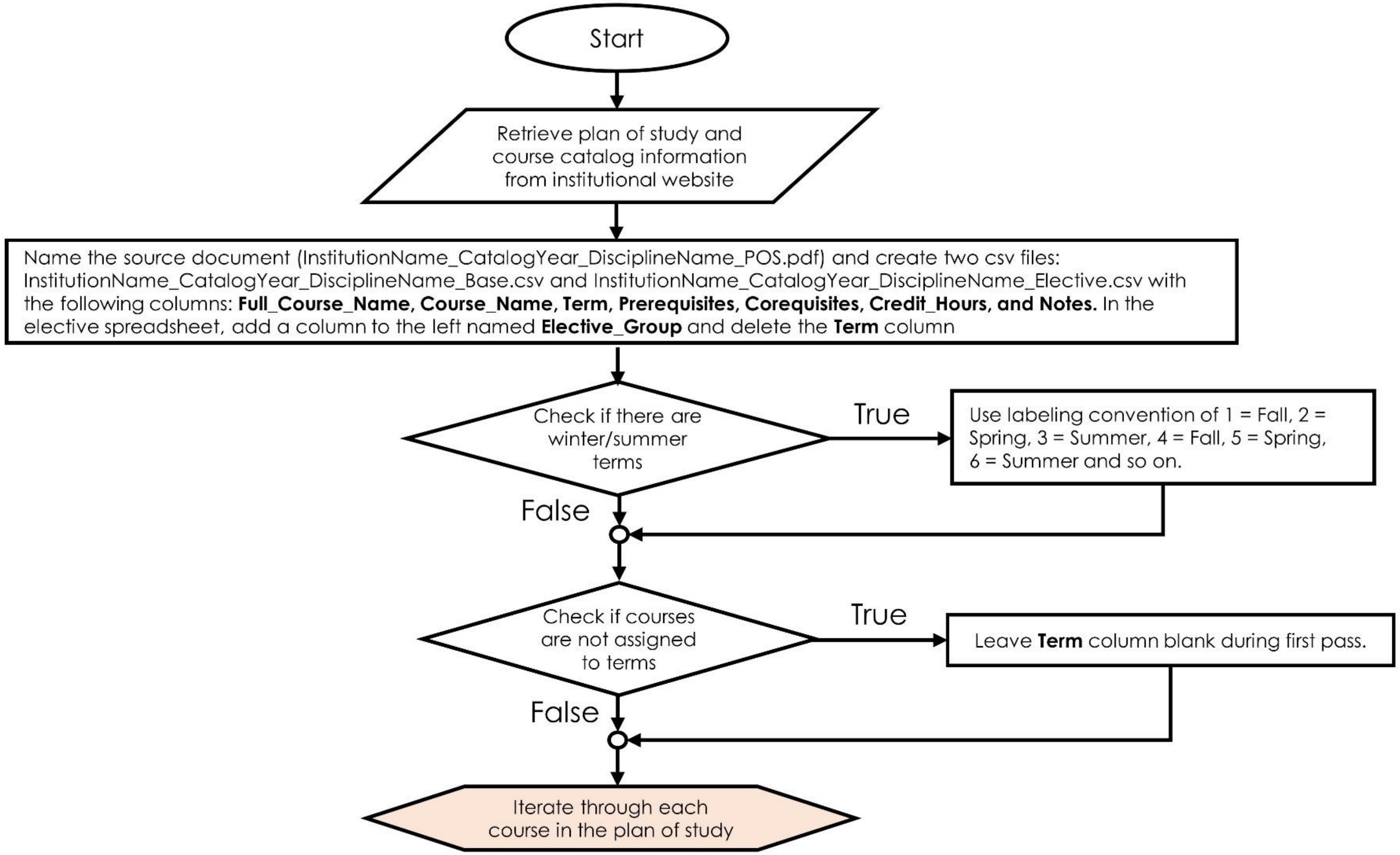

Figure 2. Pre-processing before entering data for a plan of study

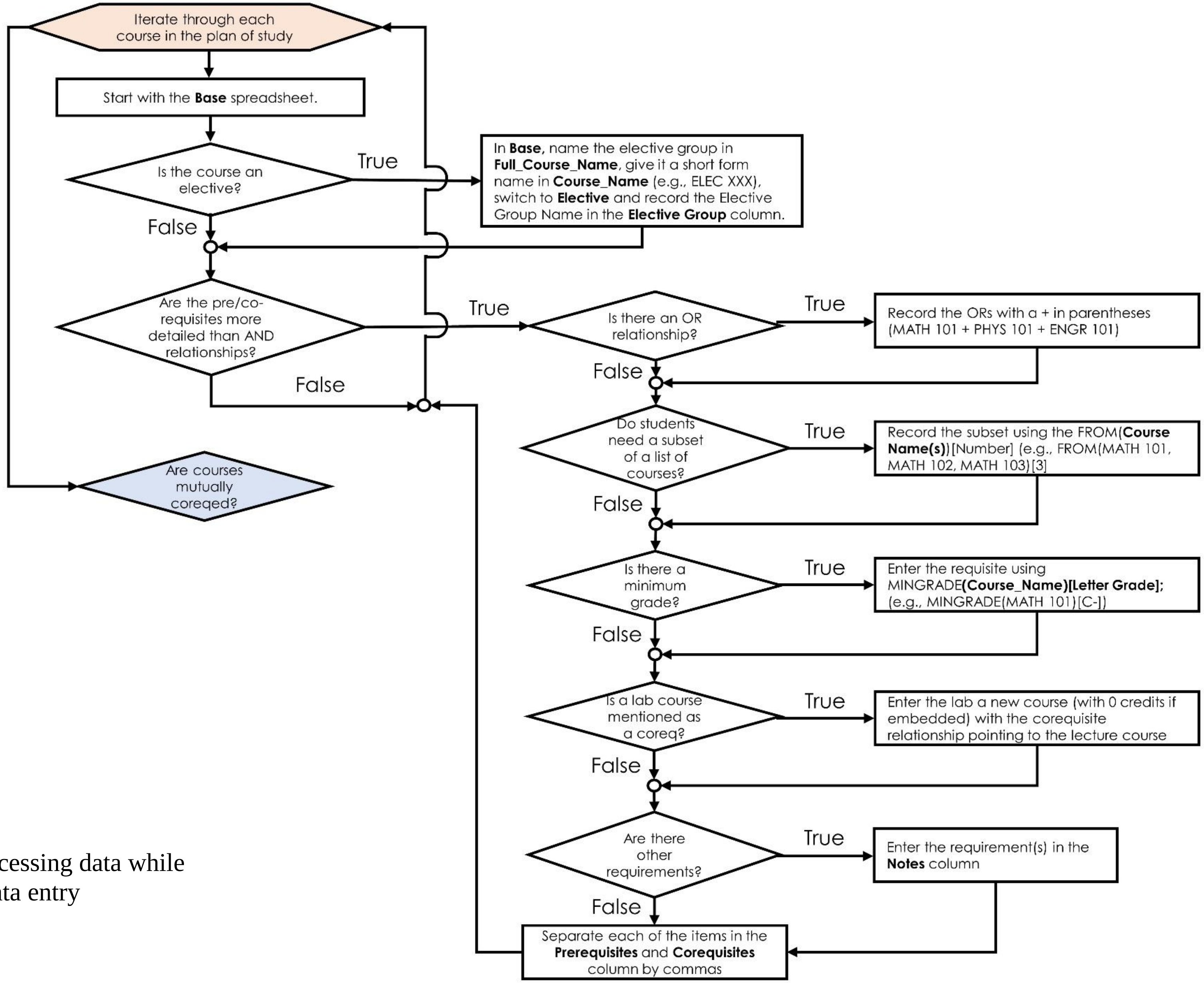

Figure 3. Processing data while performing data entry

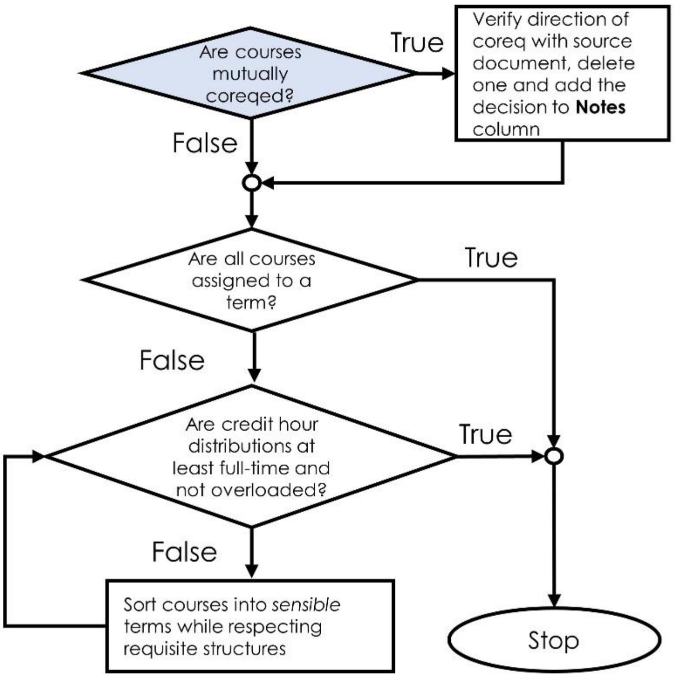

Figure 4. Post-processing after completing entry

Pre-Processing Step

Most pre-processing considerations were related to how the plans of study were organized. Therefore, questions centered on situations where the plans of study were not designed term-by-term, such as a year-by-year schedule or a simple list of courses with no prescription of when they should be taken.

Summer and Winter Terms. There were situations during our data collection process where a course was specified to be taken during a Summer or Winter term. A simple solution is to treat the extra term just like a Fall or Spring session and count the terms normally. Alternatively, and what is used in the R package in development, is the convention of 1 = Fall, 2 = Spring, 3 = Summer, 4 = Fall, 5 = Spring, 6 = Summer, and so on. Similarly, for Winter terms, 1 = Fall, 2 = Winter, and 3 = Spring. This convention makes it more visually obvious that a Summer term is incorporated and allows us to calculate metrics that incorporate limited offerings into our calculations [13], [14]. However, by labeling the terms using the explicit Summer term, we inevitably cause issues when calculating term-weighted metrics. To accommodate the labeling convention, the package automatically adjusts for the empty terms by discarding them and renumbering the terms.

To record plan of study level assumptions, notes about Summer courses like what is discussed in this section were entered into a dataset-level Word document. Upon completion of data collection, this Word document serves as a manual for the dataset allowing other researchers to understand how the individual plans of study were created – increasing research transparency.

When Plans of Study Are Not Organized by Term. Occasionally, a plan of study will not specify when a course is taken and may only provide the year or simply a list of courses. Questions about this occurred more frequently early in data collection:

- The sections are according to years, not semesters. What do I do about this?

- There's an extra summer semester in which they take 12 credits of electives. What do I do about this?
- And I've also made quite a lot of such assumptions for the 3 years of [INSTITUTION], they didn't have a semester by semester breakdown too. Where do I document every assumption I made? In the same row as the course where I made the assumption?

For this, we would suggest first entering all the required courses and prerequisite information, then organizing courses term by term based on the prerequisite relationships. That is, start with a course that has prerequisites and build out sequentially until the chain of requirements is exhausted (e.g., starting at Calculus III and working back to Calculus I). Alternatively, if the plan of study is organized by year, you can assign a temporary year column and order the courses by term within that window using prerequisite information (e.g., if year two requires Statics and Strength of Materials, Statics should go in the first term of year two because it is a prerequisite for Strength of Materials). Other courses can be unassigned a term until the more "immobile" courses are entered into the plan of study.

To form coherent plans of study, using credit hour loads to check the reasonability of a configuration is advised. If the number of credits is too small, then the student will not have full-time status. If the number of credits is too large, then the student would possibly incur overload charges. Move around less connected courses before adjusting the timing of courses in denser prerequisite structures. Some catalogs provide information on when the courses are offered, which can provide a clue to the researcher where the student is most likely to take the course in question. If not, searching the course on a website like Coursicle provides the recent semesters it was offered.

The Value Add in Term-Weighting Structural Complexity. Although we spent a good amount of effort checking the reasonability for the plan of study dataset, term numbers do not add additional information to the typical calculations in these models. Instead, they are used to create a more informative and readable visualization. However, as this research strand develops, knowing when courses occur will become more relevant for alternative metrics. For example, if one would like to analyze a curriculum from a transfer student's perspective, knowing which term they enroll would be a critical piece of information, which would be matched up with the relevant courses. Moreover, DeRocchis et al. introduced the idea of *term-weighted structural complexity* [23]. The concept involves multiplying the cruciality of a course by the term it occupies, which punishes courses that are part of dense prerequisite structures later in the curriculum. Researchers should consider exploring the weighted and unweighted structural complexity to parse the value-add of the term information.

Processing Step

Once data entry has begun, there are a few considerations necessary to consider. First is the question of electives. From a data collection perspective, the original conceptualization of Curricular Analytics suggests treating electives as generic courses without prerequisites. However, we offer an alternative perspective for enabling the study of variable pathways through a plan of study. Moreover, after discussing how electives were handled, we'll discuss salient issues with representing different types of pre- and corequisite relationships in the plan of study data.

Representing Electives. When entering courses into a plan of study, different types of electives are likely required, such as general education courses and major-specific courses. Questions about different kinds of electives were common in our chats and became more intricate as more specialized arrangements were observed:

- For [INSTITUTION] in the second semester of senior year there are many courses with no credit hours, how should I enter these?
- When it gives the student the option between choosing classes, how should we show this? [INSTITUTION] for a major in Electrical Engineering gives students the option to choose a concentration.
- I already have an option for taking 2 classes and was wondering what I should do with that.
- For a base course with no other options but varying course credits, what should we do?

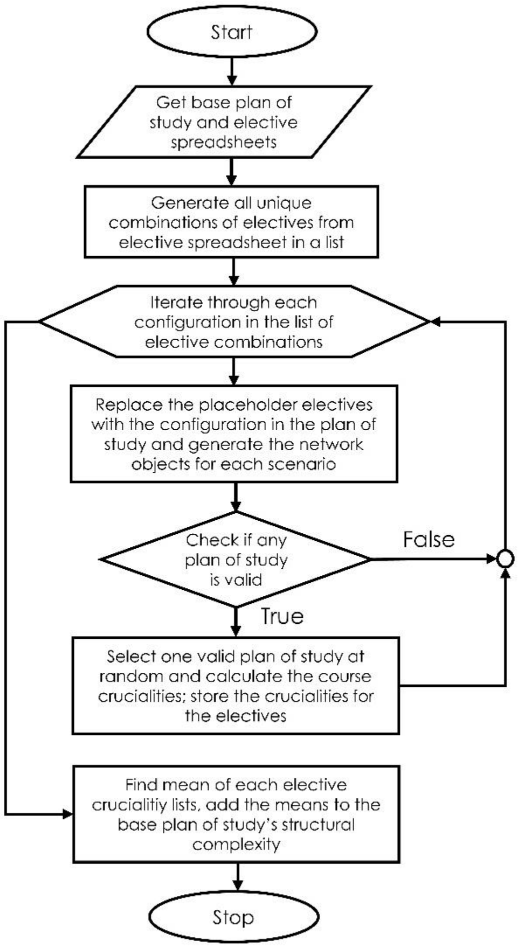

Figure 5. An algorithm for estimating the structural complexity to incorporate student choice in electives

Unfortunately, applications of curricular complexity do not explicitly address how one would incorporate student choice at such a level in the Curricular Analytics model. The most common way of representing electives is to simply leave the course unspecified and name it according to the elective type, such as “Gen Ed” or “Tech Elective.” Because the course is not specified, there is no prerequisite or corequisite information entered as well. Thus, the plan of study captures the *base* complexity as envisioned by the faculty, which does not align with how a researcher may casually discuss the complexity of the curriculum using this framework.

A more complete picture of a plan of study’s complexity can be obtained by splitting the data into two pieces, one “base” spreadsheet and one “elective” spreadsheet. The base spreadsheet will capture the name of the elective group, a short form title for the course code/number (e.g., TECH XXXX), and the term it is taken. The elective spreadsheet captures exactly what a researcher would normally enter when applying the Curricular Analytics framework, including the actual course name, course code/number, prerequisites, corequisites, credit hours, and any additional notes. When giving an elective a name, ensure it reflects the type of elective. For example, suppose there are two types of technical electives that students need to choose called Depth electives and Breadth electives. In the elective spreadsheet, we would copy the list of courses under these electives and create a new column that associates the courses with their elective group. The elective spreadsheet will serve as a pool of possible courses to pull from to estimate the expected structural complexity when we factor in student choice with electives. The algorithm for estimating the cruciality for each elective is given in Figure 5.

Using the base plan of study and elective list, we create all possible permutations of courses and store them in a list. In R, the data can be stored like so:

```
configurations <- list(c('ENGR 100','ENGR 101','ME 100'),c('ENGR 100', 'ENG
R 101', 'ME 191')) #only first two entries shown
```

Note that there will be repeated entries as individual courses are swapped out to form a new configuration, but repeated entries where a course simply switches spots with another are discarded. Once the list is generated, use a *for* loop to run through all the configurations. Check if the configuration is valid, meaning that prerequisite and corequisite relationships are not violated. If it is not valid, rearrange the courses into different elective slots on the base plan of study. If the plan of study is valid, then the cruciality of each elective is calculated. Store those in an atomic vector or data frame to keep track of the values for each configuration. Once a passable arrangement is found, we will move on to the next configuration. At the end of the loop, take the mean of the individual elective crucialities to estimate the expected cruciality. Add the estimates to the base structural complexity to yield the expected structural complexity.

The resulting adjusted value now incorporates the ability for students to choose a subset of their courses. We plan to integrate this functionality into our R package to perform the necessary analysis.

Nontrivial Prerequisites. Although many courses will have simple prerequisite structures that do not require special treatment, there are more complicated relationships that can be decomposed such that they are representable in the network. These questions were frequent in our discussions, a selection of which are below:

- This has nested "Or" options is there a good way to show this in the excel files?
- How should we show prerequisites like this given that there are "Or" + "And"? Should I just list them all and separate them with a comma?
- Also, I just saw that there was a prereq where you could take two classes from a "this or that". How should I work that into the prereq chain? [Note: “this or that” refers to a configuration where a student can pick one of two courses to satisfy a requirement.]
- MATH 160 is the first math course with MATH 161 and 261 coming in [later, and] some courses have prereqs like this: MATH 160 or MATH 161 or MATH 261, In this case, what do we write down as prereqs?
- For this it says 6 of the 7 courses, what should I do about this?
- This senior design class has a [bunch] of prereqs, and some of the are in a this or that in the POS. For example[,] GEEN 1400 is a prereq for the senior design class but you can choose between GEEN 1400 and ASEN 14000 or ECEN 1400 in semester 1. Should I keep the this or that and put the elective course in the prereq section or just choose GEEN 1400 as the only class you take?

Curricular Analytics implicitly assumes that courses have a simple “AND” configuration to describe prerequisites; for example, MATH 101 AND MATH 102 are required for ENGR 105. However, for certain courses, these configurations can involve “ORs” that are also nested within “ANDs,” such as (MATH 101 OR MATH 101H) AND PHYS 101. These are often combined with grade cut-offs, such as “Min Grade C-.” Last, some courses have prerequisite structures that offer a list of courses and require that the students have passed a subset of them using language like “at least” or “X of the following.” None of these situations are addressable in Curricular Analytics as currently defined, except that it is acknowledged on the FAQ page of the tool's website that OR relationships tend to be irrelevant [24].

To help explore whether these are indeed factors needed in curricular complexity, we introduce the following notation in Table 1 to enter the data as they are shown in course catalogs. As the framework evolves, we foresee the estimates of structural complexity to be an average of different scenarios. Our next steps involve incorporating the functionality in our R package to calculate the structural complexity for different configurations.

Table 1. Notation for prerequisite structures

Prerequisite Type	Notation	Example
AND	When a list of courses must be completed, each one is separated by a comma.	MATH 101, MATH 102
OR	When the option between prerequisites is given, we use the + symbol.	MATH 101 + MATH 102
Both Logical Connectives	Combine the comma and + notation to represent the prerequisite structure. Use parentheses to group.	(MATH 101 + MATH 102), MATH 103
Subset of Course	Use the keyword FROM and list the courses followed by the number of courses to consider in brackets.	FROM(MATH 101, MATH 102, MATH 103)[2]
Minimum Grade	Use the keyword MINGRADE, name the course and insert the minimum grade in brackets	MINGRADE(MATH 101)[C-]
Other Requirements	These can be copied verbatim into a "notes" column	"Math placement exam score of 80% or better."

We are currently experimenting with this notation, so to transform back to the original version – using purely AND connectives, we would convert all "+" to ",", remove all parentheses, and extract the courses inside the FROM and MINGRADE notation. Next, any courses that do not appear would be removed from the list of requisites. This approach would match how a researcher would conventionally enter the data. Starter R script is provided in Appendix A to accomplish the transformation.

Hidden Laboratory Coursework. Engineering curricula include laboratory-focused courses to allow students to apply principles from lecture to physical systems, and their realization in the curriculum can vary. We observed that plans of study would either have a specific laboratory course with a corequisite relationship to the lecture course, describe a laboratory experience that is mentioned by course name and number, or describe an integrated laboratory experience with no distinct course at all. Questions would emerge related to how these could be handled, for example:

- How should I show labs for a class that has the labs "baked in" with the class?
- If there is a recitation class, should I include it even if it doesn[']t show as a coreq and it has no credits?
- Some classes have recitation some don't, I wasn't sure if we wanted to call that out. The way labs for EE classes are listed is the same as above. No course name or number and no mention in the POS other than what comes up when you go to look at the prereqs and coreqs.

We found it helpful to untangle these hidden laboratory requirements by creating a course with zero credit hours with a corequisite relationship with the lecture course to unveil the additional burden on students that they create. In our view, bundling the lecture and laboratory can underestimate the true structural complexity because credit hours are not factored into the calculations at all. This convention we adopted placed all courses with labs on the same footing.

Determining Corequisite Relationships. The measures in Curricular Analytics rely on the idea of a *directed acyclic graph*, a type of network where none of the directed edges form a cycle. In other words, by following the edges from vertex to vertex, you will never be able to revisit a vertex. These curricular graphs are directed acyclic graphs by design because prerequisite and corequisite relationships naturally build on one another. It would not make sense for a course taken later in the curriculum to be a prerequisite for a course in a previous term; thus, there is no reasonable potential for a cycle to form. However, there is one exception - mutual corequisite relationships. For example, we have seen cases where a course and its laboratory section list one another as corequisites. The courses then form a cycle, which makes calculating the delay factor impossible. But the redundancy is not needed, so it is more sensible to only have one edge connecting the two.

Deciding the direction of the corequisite relationship is a nontrivial decision, which came about when the validity of the plan of study data was being evaluated. To illustrate why, consider the simplest possible configuration where the distinction is evident: Course A is a corequisite with Course B, and Course B is a prerequisite for Course C. If we have the corequisite relationship defined as Course B points to Course A, we can calculate the structural complexity to be 8. Next, by simply changing the direction of the corequisite relationship, we see the structural complexity rise to 12 – an increase of 50%! This is troubling because we did not change anything fundamental about the prerequisite relationships; the interpretation is the exact same – Courses A and B are intended to be taken together. However, the direction of the relationship has a significant impact on the course crucialities and the overall structural complexity as a result.

Why does this occur? The issue lies in the calculation of both the blocking factor and delay factor. When we change the direction of the corequisite relationship, we create a different path for requisite relationships to flow. As a result, the delay factors change for all three courses. In the first configuration, Course A does not have any paths out of the vertex but is connected to Course B, so its delay factor is 2; it also does not block any courses, so its blocking factor is 0. However, once we switch the direction of the edge, suddenly Course A becomes part of the prerequisite chain formed by Course B and Course C. This changes all the delay factors for A, B, and C to 3 and increases Course A’s blocking factor to 2.

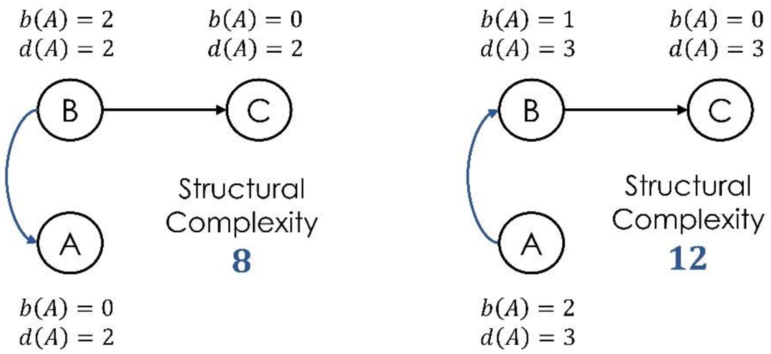

Figure 2. Illustrating the impact of changing the direction of a corequisite relationship (in blue); one change in directionality changes structural complexity by 50%; b(.) is the blocking factor and d(.) is the delay factor

The researcher must be consistent with defining the corequisite relationships to ensure the complexity values are internally consistent. If a convention for directionality is not universally applied in a dataset, then the comparability among plans of study is questionable. We could argue for either configuration, and we believe it is immaterial which orientation is chosen; it is critical that the assumption made about directionality should be reported, at least.

Another way to circumvent this decision is to ignore corequisite relationships when calculating the delay and blocking factors; one could argue that corequisite relationships are not as strict as prerequisites, and providing them with equal weight might misrepresent the complexity of a given plan of study. The advantage of ignoring the corequisites when calculating the blocking and delay factors is that the solution will be unique, but that uniqueness will come at the cost of model completeness.

Post-Processing Data

After completing data entry, it is crucial to check for accuracy in some fashion. This will involve checking for logical curricular arrangements, ensuring all courses are assigned to a term, and courseloads are reasonable as a result.

Illogical Arrangements and Inconsistencies. Occasionally, there will be instances where a configuration in the plan of study results in an illogical arrangement of prerequisites or corequisites, as evidenced in the following sample of questions:

- ***Corequisites in different terms:*** There is a course that is calling for a coreq but the course it wants as a coreq is listed under a previous year. Should I place it as a prereq instead?
- ***Prerequisites not in the plan of study because of course changes:*** For 14-15, there is as course ME 221 that prereqs ME 210: Statics of Mechanic. But, this course isn't the POS, instead there is EM 214: Statics. For 16-17, EM 215 was the prereq for ME 221. Do I assume this as a typo?
- ***Missing courses/credits:*** For the same POS, the total sum of the credits is given as 128, but is actually 129. And, there are 3 credits missing.

Notice that these are not necessarily data entry errors and are inherent to the source document. Also, though not mentioned in our chats, upon inspecting plans of study for logical arrangements, an error that occurred was accidentally listing a corequisite as a prerequisite – which would create a theoretically impossible structure.

Other issues can appear as well, so it is critical to note where potential errors have occurred. We contend these inconsistencies emerged because of our longitudinal scope, introducing situations where courses become defunct and are replaced with a new offering or sets of offerings, but this change does not become fully represented in the plan of study or catalog. During data entry or post-processing, it is suggested that researchers check for illogical arrangements, particularly with prerequisites in the same term and corequisites in different terms, against the original plan of study. If it is indeed the intended arrangement, add a note on the same line in the spreadsheet. If not, make the necessary adjustment.

Contributions to Theory in Curricular Analytics

We contend that this work advances the theory and application of Curricular Analytics in engineering education research in three distinct areas, (1) establishing a process for systematic data collection, (2) extending how researchers can account for multiple pathways in terms of electives, and (3) encouraging exploration in the impact of different curricular policies with more detailed data.

Establishing a Process of Systematic Data Collection

As of writing, there is little to no guidance for how a research team would collect data for Curricular Analytics at scale. We have illustrated through our findings in this paper that entering plan of study data across departmental, institutional, and temporal contexts carry heavy implications in terms of assumptions that must be made to achieve the desired formatting. Prerequisite and corequisite relationships are often not logically simple - that is, a set of courses that must all be taken to enroll in the course. Instead, prerequisites can have both AND and OR logical connectives (e.g., (MATH 101 and MATH 102) or (MATH 103H)), subsets of a longer list of approved courses (e.g., three of the following: ....), minimum grade requirements (e.g., MATH 101 with min C-), or a combination of all three. We are hopeful our conventions will help researchers achieve a higher level of consistency and rigor in their analyses.

Conceptualizing a Broader View on Curricular Complexity Data

Second, this work can advance how future researchers think about a broadened perspective of a plan of study's complexity. One potential critique that can be made of how data are typically entered for studies on curricular complexity is the simplification of electives down to single generic courses, washing out student agency in choosing their own pathway. Also arising from this research, and the broader literature, is student diversity resulting in varied levels of preparedness – which impacts what courses students ultimately end up taking before the prescribed coursework. Finally, we can consider how different requirements can carry unequal impacts in practice but are not treated as such in the analyses. We will briefly discuss these three points.

Curricular Complexity from Electives. For general education courses and other requirements that are mostly independent of the major-specific courses, specifying these has little impact on the structural complexity metrics. They are often labeled as "Gen Ed" or some equivalent and left without prerequisites or corequisites.

However, applying the same treatment to major-specific electives is less defensible analytically. Often upper-division courses that serve as electives have prerequisites, which are lost when they are aggregated into a generic "Tech Elective" category. Yet, incorporating the electives into the plan of study directly poses a different problem. We could specify the most common sets of electives students take, but this decision would ignore the students' agency afforded to them by the premise of these elective courses. On the other extreme, we could create a plan of study for every possible pathway, but this would be a tedious data entry task. Our proposed algorithm for estimating an average structural complexity can enable broader discussions on how students navigate the courses they have more control over and capture the curriculum's complexity more completely.

Curricular Complexity from Varied Entry Points, such as Calculus-Readiness. Building from our observations regarding nontrivial prerequisite structures, we found these most often occur with courses in the first year. Such prerequisite structures result from attempts to account for diverse offerings of mathematics courses to reflect differences in students' academic preparation. Some requirements span beyond coursework, including minimum SAT and math place scores – which we observed in our sample. When constructing the plan of study, the standard operating procedure was to only consider courses in the plan of study. However, to consider student's diverse pathways into the curriculum, researchers may consider incorporating the mathematics prerequisite structures that form the pre-calculus block of offerings. Data from Pirkey and Santiago [25] show retention statistics for students in engineering at West Virginia University, revealing that 29% percent of students were not Calculus-ready and an overwhelming majority of students in their second semester were enrolled in Trigonometry. The prevalence of non-Calculus-ready students raises additional questions about how we represent the situation in Curricular Analytics, especially considering being Calculus-ready in the first semester is a significant predictor of completing an engineering degree [26]. There are opportunities to analyze not only the expected pathway for the average first-time-in-college student but also for students who do not have the necessary math background to start in the intended mathematics course.

The notation in Table 1 can help facilitate such analyses if that is the researcher's intent. Otherwise, the notation can be converted to the standard form and analyzed normally. Only courses that appear in the list would be incorporated into the construction of the prerequisite structure in our R package.

Our suggestion in this theory paper is a two-staged data collection process that balances the desire to calculate the complexity of a curriculum using available data and incorporate student agency into the analyses. When we do not incorporate student agency, meaning we do not specify elective courses or pathways, we are not calculating the true structural complexity of the curriculum. We are calculating the expected *base* structural complexity of the curriculum *as envisioned by the faculty*. By base, we are referring to the set of courses and their prerequisites that are specified as required by the faculty in the plan of study. We add the qualifier "expected" because students have agency over the base structural complexity as well – not just electives – through transfer credit, AP credit, and exceptions. Thus, by reframing how we think and talk about structural complexity, albeit slightly, we can be more precise in its theoretical and practical applications.

Incorporating Different Requirement Types into the Analysis. Finally, through reviewing our data collection process questions, it was recognized how diverse requirements could be at different institutions. However, this diversity is not reflected in Curricular Analytics. Currently all requisites are treated equally analytically, but these requirements can have vastly different impacts. For example, the minimum grade requirement (i.e., a stricter prerequisite type than the typical relationships specified in plans of study) is undoubtedly a structural barrier that students must overcome, but it carries no analytical weight when calculating structural complexity. The minimum grade requirement is most closely associated with the blocking factor because it is an additional barrier to progress into the next set of courses, so perhaps it would be sensible to incorporate it into the metric. Alternatively, a different metric might be necessary because the minimum grade requirement can be considered a condition of the edge (i.e., the prerequisite) and not the vertex. Another possibility is to build the subnetwork of courses impacted by a minimum grade requirement and find the structural complexity of that course set. This strategy would allow one to view the set of courses impacted by a specific policy. We do not have a strong sense of which would be most appropriate without some further experimentation. However, we encourage the community to continue their own explorations with these ideas.

Conclusion

We expect that our work will prove valuable to scholars and professionals interested in carrying out systematic assessments of educational programs, particularly in conjunction with student information, to investigate issues concerning the retention of FTIC and transfer students. This paper provides some standardized procedures and data conventions to examine curricula comprehensively using network analysis. With the expansion of Curricular Analytics, we anticipate that the community will better understand the obstacles that hinder the academic progress of engineering students across the country.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. BPE- 2152441. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would also like to thank the following Undergraduate Research Assistants who helped synthesize the questions from our discussions: Elliot Setser, Autri Banerjee, and Levi Smith.

References

[1] J. Wigdahl, G. L. Heileman, A. Slim, and C. T. Abdallah, “Curricular Efficiency: What Role Does It Play in Student Success?,” presented at the 2014 ASEE Annual Conference & Exposition, Jun. 2014, p. 24.344.1-24.344.12. Accessed: Jan. 20, 2023. [Online]. Available: https://peer.asee.org/curricular-efficiency-what-role-does-it-play-in-student-success

[2] 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,” *ArXiv181109676 Phys.*, Nov. 2018, Accessed: Aug. 04, 2021. [Online]. Available: http://arxiv.org/abs/1811.09676

[3] D. Waller, “Organizational factors and engineering student persistence,” Dissertation, Purdue University, 2022. [Online]. Available: https://doi.org/10.25394/PGS.21606342.v1

[4] A. Slim, J. Kozlick, G. L. Heileman, and C. T. Abdallah, “The Complexity of University Curricula According to Course Cruciality,” in *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*, Birmingham, UK: IEEE, Jul. 2014, pp. 242–248. doi: 10.1109/CISIS.2014.34.

[5] A. Slim, “Curricular Analytics in Higher Education,” Dissertation, The University of New Mexico, 2016. Accessed: Feb. 24, 2023. [Online]. Available: https://www.proquest.com/docview/1873863748?pq-origsite=gscholar&fromopenview=true

[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 ASEE Annual Conference, Virtual, Jul. 2020, p. 15. doi: https://peer.asee.org/34363.
[7] 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.*, pp. 1–7, 2020, doi: 10.1109/TE.2020.3015090.
[8] G. L. Heileman, W. G. Thompson-Arjona, O. Abar, and H. W. Free, “Does Curricular Complexity Imply Program Quality?,” presented at the American Society for Engineering Education, Tampa, Florida, 2019, pp. 1–13.
[9] G. Heileman, M. Hickman, A. Slim, and C. Abdallah, “Characterizing the Complexity of Curricular Patterns in Engineering Programs,” in *2017 ASEE Annual Conference & Exposition Proceedings*, Columbus, Ohio: ASEE Conferences, Jun. 2017, p. 28029. doi: 10.18260/1-2--28029.
[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.*, pp. 1–30, Aug. 2020, doi: 10.1080/10668926.2020.1798303.
[11] J. Nash, L. E. Boucheron, and S. J. Stochaj, “A Correlative Analysis of Course Grades as Related to Curricular Prerequisite Structure and Inter-Class Topic Dependencies,” in *2021 IEEE Frontiers in Education Conference (FIE)*, Oct. 2021, pp. 1–5. doi: 10.1109/FIE49875.2021.9637401.
[12] 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.
[13] D. Reeping and D. Grote, “Characterizing the Curricular Complexity Faced by Transfer Students: 2+2, Vertical Transfers, and Curricular Change,” presented at the American Society for Engineering Education Annual Conference, Minneapolis, MN, 2022, pp. 1–16. doi: https://peer.asee.org/41462.
[14] D. Reeping and D. Grote, “Rethinking the Curricular Complexity Framework for Transfer Students,” presented at the ASEE Virtual Annual Conference Content Access, Virtual, 2021, pp. 1–24. doi: https://peer.asee.org/37680.
[15] B. Thorndyke, T. J. Anderson, M. Ohland, and G. Zhang, “The Creation Of The Multiple Institution Database For Investigating Engineering Longitudinal Development (Midfield),” in *2004 Annual Conference Proceedings*, Salt Lake City, Utah: ASEE Conferences, Jun. 2004, p. 9.1244.1-9.1244.10. doi: 10.18260/1-2--13092.
[16] R. Layton, R. Long, M. Ohland, M. Orr, and S. Lord, “midfieldr: Tools and methods for working with MIDFIELD data in ‘R.’” 2022. [Online]. Available: https://midfieldr.github.io/midfieldr/
[17] R. Layton, R. Long, M. Ohland, M. Orr, and S. Lord, “midfielddata: MIDFIELD data sample.” 2022. [Online]. Available: https://midfieldr.github.io/midfielddata/
[18] M. W. Ohland, R. A. Long, S. M. Lord, M. K. Orr, and C. E. Brawner, “Expanding Access to and Participation in the Multiple Institution Database for Investigating Engineering Longitudinal Development,” presented at the 2016 ASEE Annual Conference & Exposition, Jun. 2016. Accessed: Feb. 12, 2023. [Online]. Available: https://peer.asee.org/expanding-access-to-and-participation-in-the-multiple-institution-database-for-investigating-engineering-longitudinal-development
[19] J. Walther, N. W. Sochacka, and N. N. Kellam, “Quality in Interpretive Engineering Education Research: Reflections on an Example Study: Quality in Interpretive Engineering Education Research,” *J. Eng. Educ.*, vol. 102, no. 4, pp. 626–659, Oct. 2013, doi: 10.1002/jee.20029.
[20] J. Saldaña, *The coding manual for qualitative researchers*, 3rd ed. Sage, 2016.
[21] A. Strauss and B. Glaser, *The discovery of grounded theory: Strategies for qualitative research*. Sociology Press, 1967.
[22] S. Spall, “Peer Debriefing in Qualitative Research: Emerging Operational Models,” *Qual. Inq.*, vol. 4, no. 2, pp. 280–292, Jun. 1998, doi: 10.1177/107780049800400208.

[23] A. M. DeRocchis, L. E. Boucheron, M. Garcia, and S. J. Stochaj, “Curricular Complexity of Student Schedules Compared to a Canonical Degree Roadmap,” in *2021 IEEE Frontiers in Education Conference (FIE)*, Oct. 2021, pp. 1–5. doi: 10.1109/FIE49875.2021.9637443.
[24] “Curricular Analytics.” https://curricularanalytics.org/faq (accessed Feb. 16, 2023).
[25] A. C. Pirkey and L. Santiago, “Understanding the Educational Path of Non-Calculus-Ready Students in Engineering,” presented at the 2021 ASEE Virtual Annual Conference Content Access, Virtual, 2021, pp. 1–7. doi: https://peer.asee.org/37967.
[26] B. Bowen, R. Hall, and J. Ernst, “Calculus eligibility as an at-risk predictor for degree completion in undergraduate engineering,” *Technol. Interface Int. J.*, vol. 18, no. 1, pp. 74–80.

Appendix A: R Script to Convert Data Using Our Introduced Conventions to the Original Form for Curricular Analytics

```r
#This function takes in the columns for the pre- and corequisites
#and removes the supplementary information in our suggested notation
#such that it will work with the conventional functions for
#Curricular Analytics
ConvertToOriginalNotation <- function(RequisiteStructure) {
  #Replace all ORs (+) with ANDs (,)
  RequisiteStructure <- gsub("\\+", ",",RequisiteStructure)
  #Remove the parentheses
  RequisiteStructure <- gsub("[()]", "", RequisiteStructure)
  #Begins removing the notation for MINGRADE and FROM.
  RequisiteStructure <- gsub("FROM|MINGRADE", "", RequisiteStructure
)
  #Remove notation for MINGRADE and FROM.
  RequisiteStructure <- gsub("\\[.*?\\]", "", RequisiteStructure)
  #Return the courses with the original notation.
  return(RequisiteStructure)
}

#Apply the function to each of the courses in the prerequisite and
# corequisite columns.
ConvertedRequisites <- sapply(RequisiteStructures,
                              ConvertToOriginalNotation)
```