A Network-based Analysis of Technology-driven and Load-driven Constraints in Production Data

by

Serhat Kosif

a thesis for conferral of a Master of Science in Data Engineering

Prof. Dr. Marc-Thorsten Hütt
First reviewer

Prof. Dr. Yılmaz Uygun
Second reviewer

Date of Submission: 22/07/2021

JACOBS UNIVERSITY

Statutory Declaration

Family Name, Given/First Name	Kosif, Serhat
Matriculation number	30003342
What kind of thesis are you submitting: Bachelor-, Master- or PhD-Thesis	Master-Thesis

English: Declaration of Authorship

I hereby declare that the thesis submitted was created and written solely by myself without any external support. Any sources, direct or indirect, are marked as such. I am aware of the fact that the contents of the thesis in digital form may be revised with regard to usage of unauthorized aid as well as whether the whole or parts of it may be identified as plagiarism. I do agree my work to be entered into a database for it to be compared with existing sources, where it will remain in order to enable further comparisons with future theses. This does not grant any rights of reproduction and usage, however.

This document was neither presented to any other examination board nor has it been published.

German: Erklärung der Autorenschaft (Urheberschaft)

Ich erkläre hiermit, dass die vorliegende Arbeit ohne fremde Hilfe ausschließlich von mir erstellt und geschrieben worden ist. Jedwede verwendeten Quellen, direkter oder indirekter Art, sind als solche kenntlich gemacht worden. Mir ist die Tatsache bewusst, dass der Inhalt der Thesis in digitaler Form geprüft werden kann im Hinblick darauf, ob es sich ganz oder in Teilen um ein Plagiat handelt. Ich bin damit einverstanden, dass meine Arbeit in einer Datenbank eingegeben werden kann, um mit bereits bestehenden Quellen verglichen zu werden und dort auch verbleibt, um mit zukünftigen Arbeiten verglichen werden zu können. Dies berechtigt jedoch nicht zur Verwendung oder Vervielfältigung.

Diese Arbeit wurde noch keiner anderen Prüfungsbehörde vorgelegt noch wurde sie bisher veröffentlicht.

22/07/2021 .....

Date, Signature

Abstract

Constraints lead to statistical patterns in data. The initial step of this master thesis work is to quantify the characteristics of two hypothetical types of constraints in industrial production: technology-driven constraints and load-driven constraints. That was achieved by analyzing the statistical properties of association networks over time in large data sets from steel manufacturing. Based on these results, an abstract theoretical framework was developed to understand better the connection between each type of constraint and its statistical patterns.

Acknowledgements

I would like to acknowledge Professor Dr Marc-Thorsten Hütt, principal investigator in the SMS group project, for motivating this project and the valuable information presented in our meetings. Many of the ideas presented herein were taken from those meeting conversations. I also wish to thank Daniel Christopher Merten for reviewing this document and ripping it to shreds, as every good advisor should do to a thesis draft. Lastly, I wish to thank Dr Atilla Özgür for providing me access to the data source that the complete thesis work is established.

Contents

List of Symbols and Acronyms

$\mathbb{R}$ Set of Real Numbers

$\mathbb{Z}$ Set of Integers

$\Delta\mathbf{Q}$ Additional Contribution for Modularity

$\boldsymbol{\mu}$ Expectation Value (Mean) for Modularity in Randomised Graphs

σ Standard Deviation of the Modularity Values in Randomised Graphs

$\mathbf{A}$ Adjacency Matrix

$\mathbf{A}^m$ Metabolite-centric Adjacency Matrix

$\mathbf{A}^r$ Reaction-centric Adjacency Matrix

ATP Adenosine Triphosphate Production

CCM Continuous Casting Machine

CGL Continuous Galvanizing Line

CSP Compact Strip Production

D An Arbitrary Manufacturing Data Set

f(x) Normalisation Function

FBA Flux Balance Analysis

FBS Fixed Bucket Sized

FSS Fixed Step Sized

GCN Genome-scale Cellular Network

k Network Vertex Degree

kg Kilogram

m Network Edge Number

mm millimetre

NM-d Null Model conserving degrees sequence

NM-m Null Model conserving degrees sequence & graph modules

O Objective Function Coefficients Array

OR Operations Research

PLTCM Pickling Line & Tandem Cold Mill

PPD Product Portfolio Diversification

Q Modularity

RU Resource Utilisation

S Stoichiometric Matrix

s Divided Network Group

SQL Structured Query Language

T-shape Standard Deviation of the Error Bars

V Flux Vector

$\mathbf{V}^b$ Constrained Flux List

$\mathbf{V}^e$ Deleted Flux List

z Standard Score Formula

z-score Standard Score

List of Tables

List of Figures

List of Equations

1 Introduction

1.1 Background Information and Motivation

An alloy of iron and carbon, steel has notable durability with superior mechanical characteristics. Based on the evolving ability of its microstructures [1, 2], steel compositions can be obtained in a wide range, and they can be recycled without loss of property. Those make steel an excellent material to meet the ever-changing requirements of our contemporary society.

A steel manufacturing facility has a complex structure consisting of various production lines and can process a set of products successively. Fundamental steel manufacturing steps with respective production units are shown in Fig. 1.1 [3]. Raw materials are melted in the blast, electric arc or basic oxygen furnaces to obtain liquid iron. In the next step, liquid steel alloy is sent to the continuous casting line. It is poured into a mould cavity where it starts to cool. Right after, it is treated in a secondary cooling process with water sprays until it solidifies. A further production line involves a rolling process, which can be performed in two different modes: hot rolling and cold rolling. It allows obtaining the desired mechanical properties of steel, uniform thickness, a control on width dimension, and converting material to a flat and rectangular slab (the so-called production event), a semi-finished steel product. The product is heated in a reheat furnace before it's subjected to high pressures in the hot rolling unit. Before the cold rolling process, the product is treated with pickling to remove rust and impurities on the slab surface, allowing easier work on the material. The cold rolling unit improves surface finish and flatness and allows to adjust metal work hardening. After the rolling process, slabs are converted into compact coils featuring high lengths unless they will not be sent to further units on the continuous production line like a galvanising line. It is a protective zinc coating application on the surface to improve corrosion resistance.

The whole continuous production process is scheduled in a sequencing fashion, and production events are grouped into batches (the so-called production sequences). Sequences are arranged by the common properties of events and defined priorities for operation efficiency. Various constraints arise in technical, logistic, physical and chemical aspects [4, 5] due to different machines and production lines; having efficient sequence planning is necessary to cope.

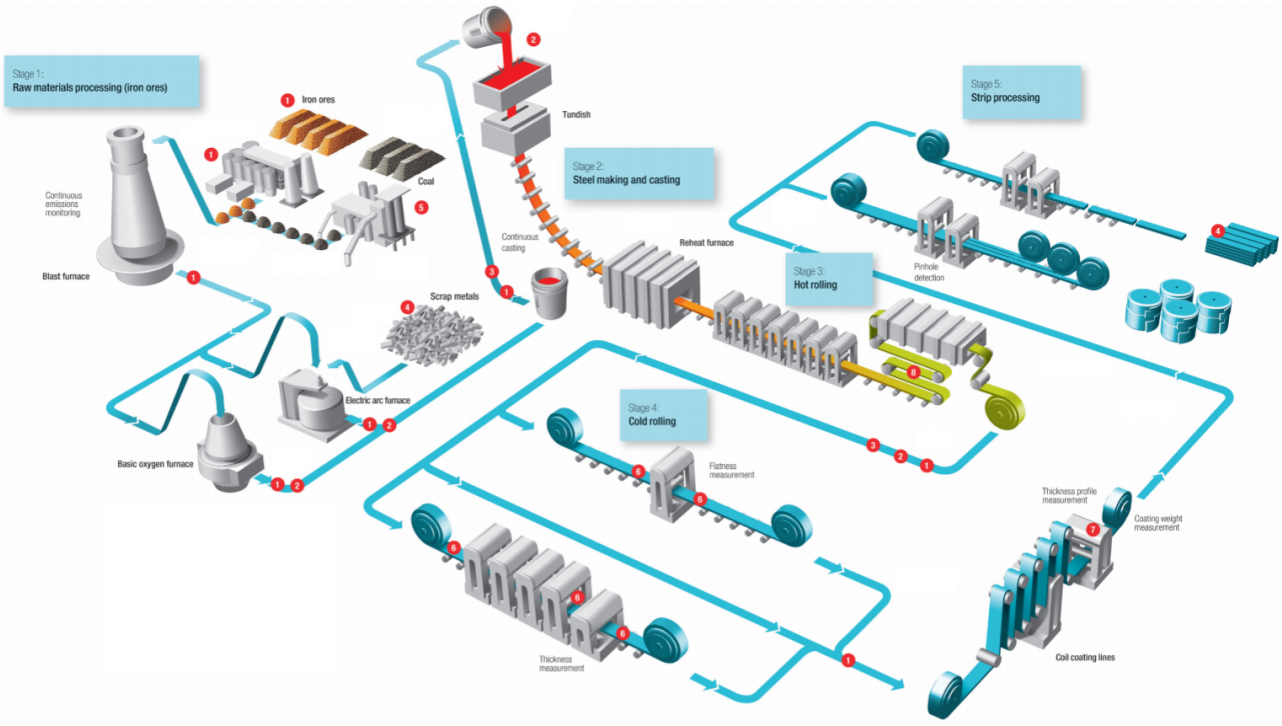


Figure 1.1: Steel Manufacturing Steps.

The above-explained processes and workstations can be arranged as a variety of integrated solutions based on the requirements of different demands or facility organisations. Table 1.1 shows four production lines that the SMS group supplies [6].

Table 1.1: Compact Production Lines with Various Machine Modules.

No	Production Unit / Line	Description
1	Continuous Casting Machine (CCM)	Steel cools, passing through the mould cavity and solidifies after water spraying.
2	Compact Strip Production (CSP)	Compact plant including CCM, reheating furnace, hot rolling unit and strip processing unit.
3	Pickling Line & Tandem Cold Mill (PLTCM)	Compact plant including a turbulence pickling section and a tandem mill.
4	Continuous Galvanizing Line (CGL)	Application of protective zinc coating on the steel surface to improve corrosion resistance.

Steel manufacturing factory systems are automated with combined computer control and digital information. The sensory information received from production line machines concerning production events is stored as data within a server to be incorporated with planning progress to provide functionality, adaptability and effective resource allocation in manufacturing processes [7]. A data collection of production events from a steel manufacturer was pulled from the SMS group database to investigate. The queries in Structured Query Language (SQL) were generated to find the production events across 2–3 years of production work completed in the production lines, introduced in Table 1.1. The SQL queries are attached as supplementary materials, S1, S2, and S3. The query attributes and resulting attribute values are anonymous. Further details for the data collection and cleaning steps are given in Subsection 3.1.1.

After manipulation and cleaning steps were performed, CCM, CSP, PLTCM, and CGL data sets have the number of events, 347418, 205496, 59604, and 27147, respectively. The decreasing number of events through the data sets shows that the output of a production line is not always an input for the next in line and might be excluded from downstream production lines, as previously mentioned. As shown in Fig. 1.2, the number of events handled decreases with a slightly reduced width variety from CSP to CGL. Moreover, the production events in PLTCM and CGL have a wide diversity and precision in a narrow range for the thickness feature. In contrast, the events in CSP have a precise unimodal distribution gathered around a single value in a broad range of the same feature values.

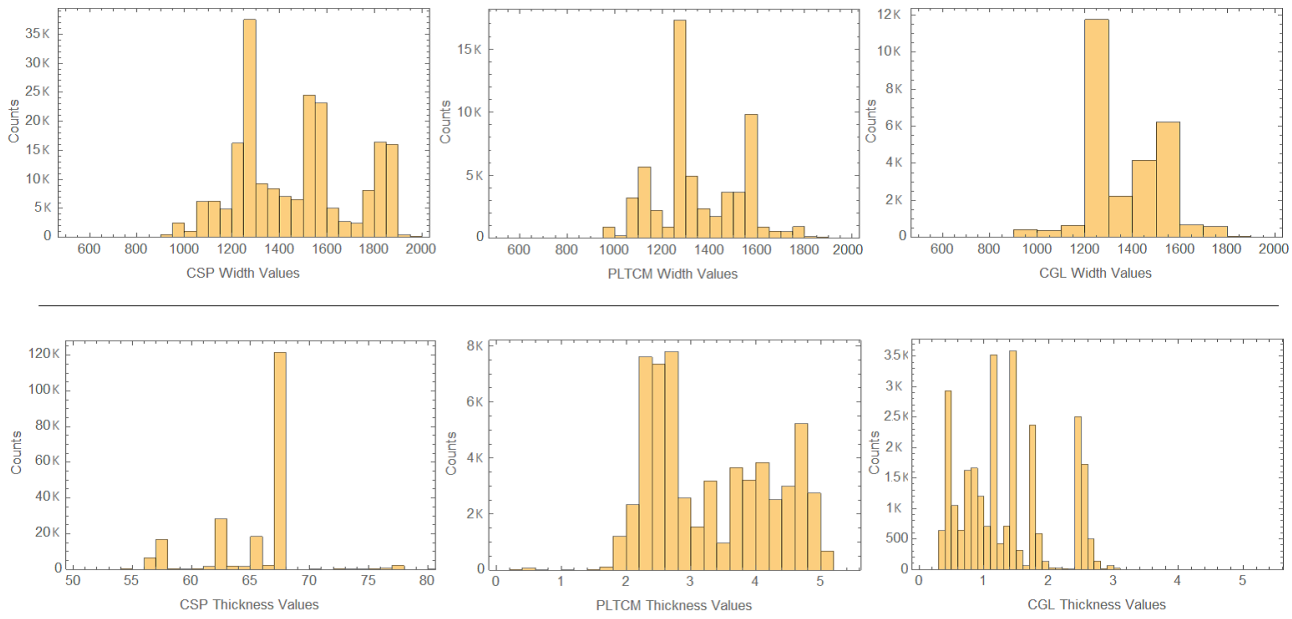


Figure 1.2: Diversity in Event Features for Different Production Lines.

Based on the summary of production data sets given above, the features of the production events lose diversity and gain more precision as going further from CCM and CSP to PLTCM and CGL. In other words, a more generic production plan gives its place to a plan with more speciality among the production events.

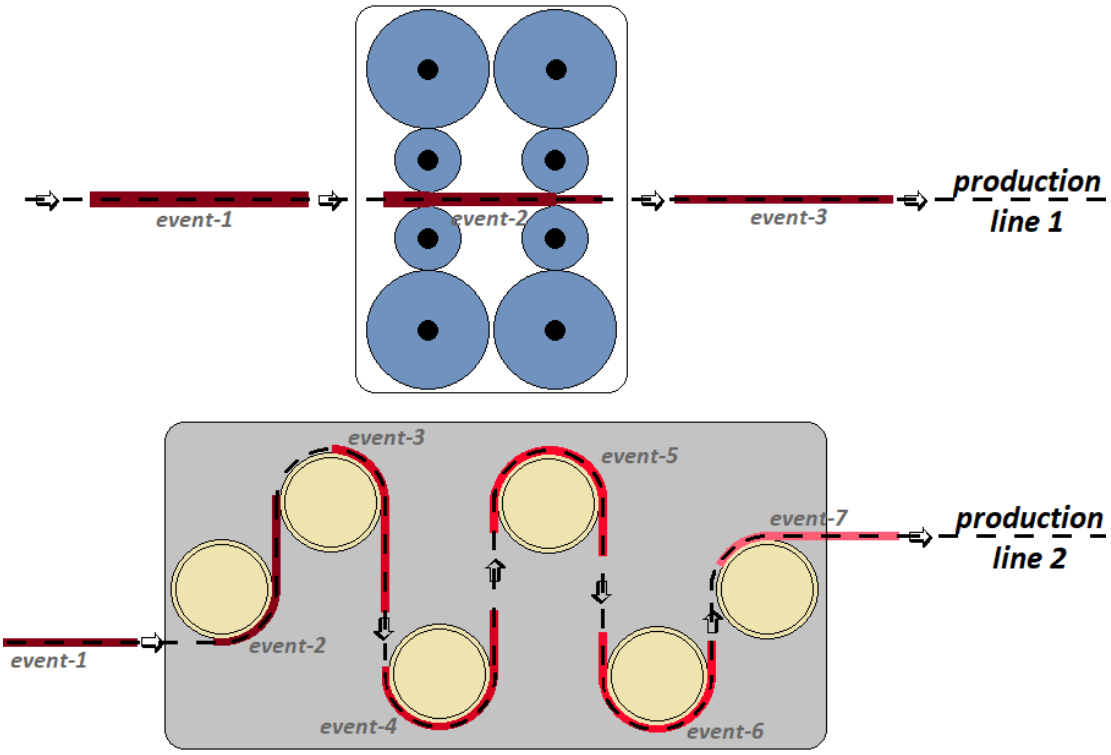


Figure 1.3: Different Categories of Production Events Handling. The hot rolling mill positioned in production line 1 treats only one slab at a time. The unit in production line 2, pickling tank full of acid coloured in grey, treats the metal surface of more than one slab at a time.

In addition to the diversity of the products, one can also see the handling operation differs between those production lines. Fig. 1.3 illustrates alternative handling categories for the production events in two separate production lines. The rolling machine in production line 1 can handle one event at a time. New rollers yet installed adapt to work initially, whereas the already adapted rollers can perform better on small width dimensions until they start to underperform and accordingly be replaced. Therefore, the events need to be arranged in a specific order based on their widths to get the best performance [5]. That brings a technical limitation. In contrast, the pickling tank positioned in production line 2 performs continuous handling of more than one production event simultaneously, yet with a maximum volume of events [8], bringing a load limitation.

1.2 Research Objective and Plan

We argue that the limitations introduced in Fig. 1.3 are fundamentally different from each other. We call those as technology-driven constraints, which arise from the handling type in production line 1 and the others as load-driven constraints, which come from the handling type in production line 2. Also, we claim that the technology-driven constraints are effective in CCM and CSP since the machines belonging to those production lines handle the events one by one successively. In contrast, the load-driven constraints are binding in PLTCM and CGL since the handling is mainly performed up to certain volumes of events at a time in those production lines.

We ask if those two different constraints can be discriminated with a formal definition, leading us to observe related functional consequences. Introducing alternative binning schemes to group production events based on various machine limitations would quantify two hypothetical types of constraints introduced above. It will allow us to investigate if they impact how the production system behaves. Further details related to binning schemes are given in Subsection 2.1.2.

As the first step, we constructed an analysis pipeline on steel production data concerning the identified binning schemes, which would provide us with the statistical impacts to discriminate the two types of constraints. We used the data sets belonging to the production lines introduced previously and analysed them in time intervals to observe the statistical impacts.

As the second step, we developed an abstract theoretical framework inspired by a novel mathematical modelling approach used by metabolic engineers to simulate microbial metabolisms belonging to various living organisms. We ran experimental simulations among the framework that would help understand the difference between those constraints mechanistically and their statistical patterns.

The two steps as mentioned above constitutes our Operations Research (OR) model, which combines steel production events analysis and constraint-based simulation framework. The efficacy of the OR model comes from structuring a standard data format and a shared analysis logic that compares the results from steel production data and simulation data.

2 Methods

This chapter introduces the proposed concepts used in our OR model as association networks, binning methods, network metrics analysis, and an optimisation scheme, Flux Balance Analysis (FBA) for the simulation framework.

In the coming section, we introduce the integrated techniques to create non-random topological features in the production data association networks that come from different types of constraints.

Afterwards, we introduce the Flux Balance Analysis, which consists of a linear set of fluxes of material flows. The goal is to find the pattern of material flow that maximises the output. The advantage of this approach: the notion of constraints is already present in that framework.

2.1 Steel Production Events

2.1.1 Association Networks

Beyond a simple network graph representation of a historical production data, the formation of association rules networks is an insightful graph-based framework combining the tools: association rules and complex networks, as Merten et al. (2020) performed in their article [9]. Their pipeline considers sequentially revealed events of a data set. It outputs a graph demonstrating the non-random occurrence of specific events together among the complete set that took place consecutively in the production period.

Assume we have an arbitrarily created production data set with chronological order, D, consists of k sequences and n events with Feature-A values and sequence id's included as given in Table 2.1.

By looking at such a data set, one can say the events with Feature-A values: 890, 850, 650, 745, 795 or 540, 520, 630, 610 are positioned in common sequences and close to each other; thus, they are produced together and likely occur in the identical sequences. As a further argument, the conclusion mentioned above is probably a deliberate planning choice based on the related constraints acting on the manufacturing process performance. However, extracting such implicit

Table 2.1: Arbitrary Production Data Set D.

Event ID	Feature-A	Sequence ID
1	280	1
2	250	1
3	890	2
4	850	2
5	650	2
6	745	2
7	795	2
8	150	3
⋮	⋮	⋮
n-4	940	k-1
n-3	540	k
n-2	520	k
n-1	630	k
n	610	k

knowledge is not a simple task for large and complicated steel production data. For example, such a data set may consist of more than $300,000$ events and is likely to have various events aggregated randomly in its large sequence groups.

We extract the association rule from the set of production sequences to distinguish statistically unexpected occurrences from the non-random ones in production sequences and assess the complexity of production patterns. The association rule measure, known as Lift, was picked with a similar approach as Merten et al. (2020) applied in their article [9]. It was calculated for every possible pairwise subset of Feature-A values belonging to the events in identical production sequences. The Lift can be computed as the ratio of pair items joint probability divided by the multiplication of each item's marginal probability as

$$Lift(A \leftrightarrow B) = \frac{P(A,B)}{P(A) * P(B)}. \tag{2.1}$$

In the case of $Lift(A \leftrightarrow B) > 1$, B occurs likely if A occurs while $Lift(A \leftrightarrow B) < 1$, B unlikely occurs if A occurs. Indication of random and non-random co-occurrences as 0 and 1 in an adjacency matrix will provide the data structure to form an association network, as shown in Fig. 2.1.

Events	1	2	3	...	n-1	n
1	0	1	0	...	0	0
2	1	0	0	...	0	0
3	0	0	0	...	0	0
⋮	⋮	⋮	⋮	⋱	⋮	⋮
n-1	0	0	0	...	0	1
n	0	0	0	...	1	0

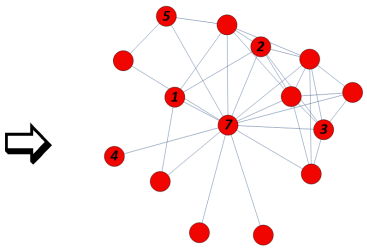


Figure 2.1: An Arbitrary Representation for Adjacency Matrix and its Graph.

2.1.2 Binning Schemes

The data set D events can be labelled with a typical value interval (the so-called binning size) for the Feature-A values with a slight difference. Binning generation can be performed in alternative ways, allowing us to put the hypothetically created constraints into practice.

Say that we do the Feature-A values labelling with a typical binning size, in our case, 99, so that all of the events in D must match the corresponding Fixed Step Sized (FSS) interval, as shown in Table 2.2.

Table 2.2: Data Set D with FSS Bin Size Labels.

Event ID	Feature-A	FSS Bins	Sequence ID
1	280	200-299	1
2	250	200-299	1
3	890	800-899	2
4	850	800-899	2
⋮	⋮	⋮	⋮
n-2	520	500-599	k
n-1	630	600-699	k
n	610	600-699	k

An alternative way of label generation is to create bins with equal event counts per bin among the complete data set; Fixed Bucket Sized (FBS) labelling is shown in Table 2.3. The alternative binning generation methods mentioned above let us derive two distinguished approaches to construct association networks. The first one is the FSS Network approach; it has graph nodes as binning groups with equal bin sizes. Manipulation of binning size allows us to aggregate events in different network nodes. The FBS Network approach is the second one where the network nodes are binning groups with an equal number

Table 2.3: Data Set D with FBS Bin Size Labels.

Event ID	Feature-A	FBS Bins	Sequence ID
1	280	200-599	1
2	250	200-599	1
3	890	630-899	2
4	850	630-899	2
⋮	⋮	⋮	⋮
n-2	520	200-599	k
n-1	630	630-899	k
n	610	600-629	k

of events per bin. Defining a typical bucket size for the network nodes results in arbitrary interval boundaries for each node, and it allows to control their population.

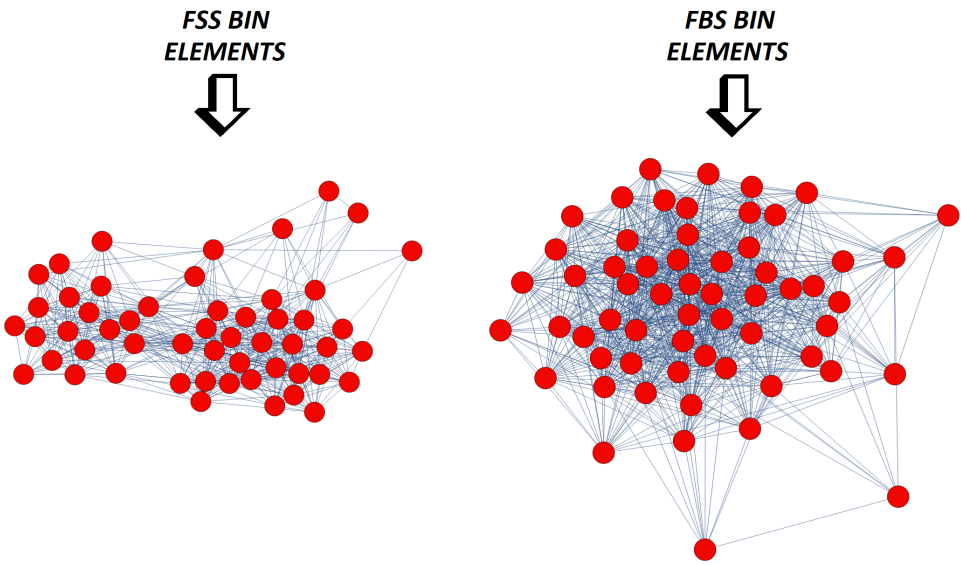


Figure 2.2: Graph Results for Two Different Network Approaches.

Constructing FSS and FBS networks for the production events concerning technology-driven constraints and load-driven constraints underlie the developed hypothesis of this thesis work: Non-random features of the association networks derived from these two approaches.

2.1.3 Network Metrics Analysis

As explained in the previous subsection, one can label a data set differently and generate identical graphs with FSS Network and FBS Network approaches. We argue that resultant graphs have various motifs which are non-trivial and emerge from the statistical patterns in data. In further steps in this subsection, we review a well-known network measure: modularity and statistical techniques of randomness control to be integrated into our analysis pipeline.

Modularity Measure

The variety of textures arises from how the nodes are clustered within their neighbourhood or having different degree values. The degree is a network metric that quantifies one node's links (or edges) to the other nodes [10]. The degree distribution of the network gives an idea about the connectivity patterns within the network. It allows us to distinguish the nodes with a high degree from the nodes with a low degree.

Identification of tightly connected node groups is a way of quantifying community structure in networks [11]. Communities (the so-called modules) are groups of nodes that probably play similar roles within the graph [12]. Modularity is a network measure for community detection and quantifies the strength of community structure in that specific network. It is a way to express the network characteristics.

Newman (2006) formulated modularity in his article as

$$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j, \tag{2.2}$$

where the network graph has an m number of edges, and A_{ij} is the number of edges between vertices i and j. A_{ij} is the element of the adjacency matrix introduced in Fig. 2.1. It can be 0 or 1. k_i and k_j are the vertex degrees, and $k_i k_j / 2m$ is the expected number of edges between i and j if edges are randomly placed. s_i and s_j are the divided network groups. They are equal to 1 if i and j belong to the same group and 0 otherwise. Eq.(2.2) is used to separate the network into two communities only; however, many networks may contain more than two communities. Therefore, a repeated division into two is adapted: dividing the network into two graphs, then the two sub-graphs further divided into two only if that would maximise Q. After first partitioning, the edges falling between the further divided sub-graphs are neglected, leading to a wrong maximisation quantity. For this reason, the author introduced the additional contribution ΔQ. [13]

The formulation given in Eq.(2.2) was used in this thesis work to calculate the modularity of the association networks. Since the results obtained with the combination of Q and ΔQ do not significantly differ from the results obtained only using Q, the modularity calculations in this work were performed with the latter one to lower the computation timing.

Randomness Control Concerning Different Null Models

As Eq. (2.2) gives a clue, one can measure a real network modularity quality by comparing it with the community structure in a random graph [14]. The distribution of degrees in random graphs is highly homogeneous, and they do not reveal a significant level of order or organisation [12]. Various sophisticated random graphs (the so-called null models) can be generated from the original network graph by keeping some of its structural properties the same [15, 9, 12, 16].

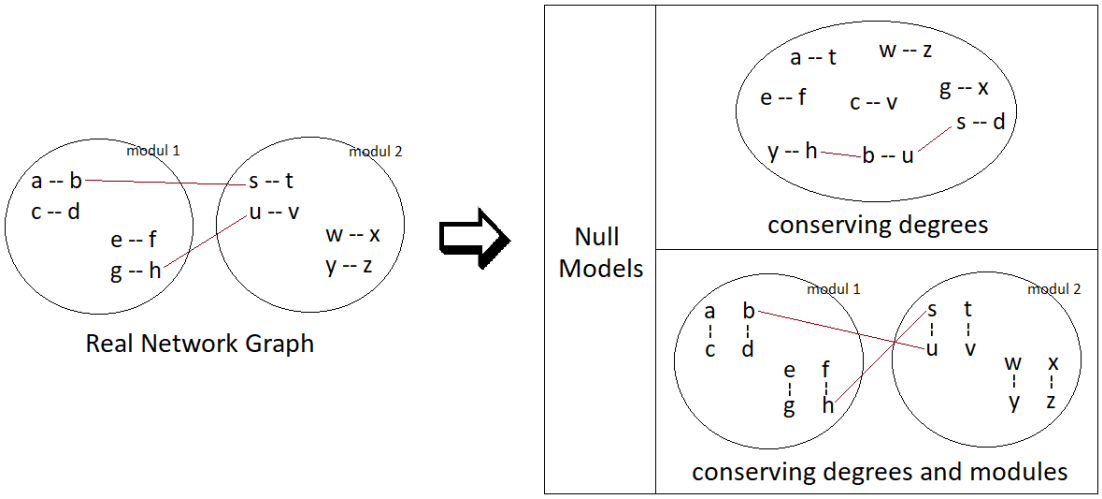


Figure 2.3: Formation of Different Null Models.

Our analysis pipeline considers two types of randomised graphs for our association networks: Null Model conserving degrees sequence (NM-d) and Null Model conserving degrees sequence & graph modules (NM-m), as shown in Fig. 2.3. In NM-d, all edges that belong to the real network are shuffled in a pairwise fashion by keeping the original degrees sequence which allows conserving any possible skewed degree distribution in the real network [15, 17, 12]. In NM-m, intra-edges and inter-edges among modules are shuffled separately by preserving the original degrees sequence [17]. We should emphasise an essential detail in our design decision that might affect the results; NM-m keeps inter-edges from different module pairs together while the shuffling process, even if there

are more than two modules in the real network. However, some module pairs might be strongly interconnected in most realistic situations, while the others are almost not linked to each other.

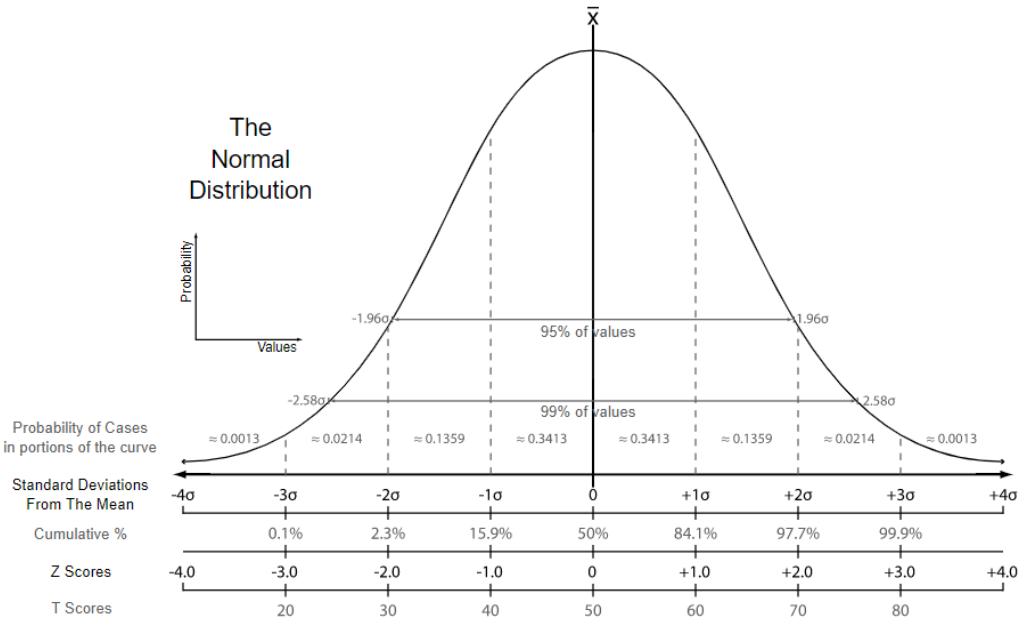


Figure 2.4: Chart Comparing the Various Grading Methods in A Normal Distribution.

One thousand random graphs concerning the respective null model constraints are created, and their modularity values are computed to compare with the real network. The histogram of resulted modularity values converges to a normal distribution like the one shown in Fig. 2.4 [18]. One can quantify the real network randomness in the context of the respective null model by computing the Standard Score (the so-called z-score), z, as

$$z = \frac{Q - \mu}{\sigma}. \tag{2.3}$$

Q is the modularity value for the real network; μ is the expectation value (mean) of Q in the set of 1000 randomised graphs. σ is the standard deviation of Q in the randomised graphs.

z is the number of standard deviations by which the real network modularity value is below or above the expected value, μ. The z-score lower than 1 or higher than -1 indicate that the real network is incidental; the z-score between 1 and 2 or between -2 and -1 suggest that the real network is close to random

characteristics. In contrast, the z-score greater than 2 or less than -2 indicate a significant deviation from randomness.

NM-d is the null model that gives information about the modularity since it destroys the modules in the real network while randomising it. NM-m is essentially the control null model to detect strange effects and whether it is meaningful to discuss modularity. Comparing a real modular network with NM-m random graphs will lead the z-score to zero, no matter the actual modularity value. If the z-score using NM-m is drastically away from zero, then the type of modularity in the real network graphs are somewhat different and very complicated.

In some networks, small groups of nodes organised by following a hierarchical rule [19] form large groups displaying a high degree of clustering while the degree distribution follows a power law [20]. That hierarchical organisation of nodes creates a nested modularity structure in the networks, having modules within modules. That type of organisation is observed in several real networks like the World wide web, the Internet at the domain level, actor-network [20], macaque & cat cortical systems [21] and the Escherichia coli metabolic network [22]. We assume that the real network has a complicated structure or hierarchical organisation since the randomising scheme would destroy the internal modularity of graph modules if the z-score concerning NM-m is lower than -2 or higher than 2.

Table 2.4 summarises possible network structures, including our assumptions for the respective z-score intervals under the effect of null model choice.

Table 2.4: Expected Network Structures Concerning Different Null Models.

Real Network Z-score	Conserving Degrees Sequence (NM-d)	Conserving Degrees Sequence and Modules (NM-m)
$z \leq -2$	nonrandom, non-modular	hierarchical
$-2 < z < 2$	random, non-modular	simple
$z \geq 2$	nonrandom, modular	hierarchical

2.2 Simulation Events

The Genome-scale Cellular Networks (GCN) are necessary tools used by metabolic engineers on model design, theoretical and computational analysis to understand how the biological system of microbial organisms works [23]. In addition, integrated network theory tools expand the feasible space for analysis techniques in further work steps. As an initial step, one can construct a network showing interactions between metabolites, intermediate or end products and metabolic reactions for an organism.

The set of rules for the organism can be represented in a compact form by an m-by-r matrix formulation as

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1r} \\ s_{21} & s_{22} & \dots & s_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \dots & s_{mr} \end{bmatrix} = (s_{ij}) \in \mathbb{Z}^{mxr}. \tag{2.4}$$

The matrix S is called stoichiometric matrix, its column elements represent reactions that play a role in the chemical transformations, and its row elements represent metabolites. S also contains direction information for the related metabolite-reaction element in the matrix with positive or negative signs. [24]

Having transpose of S will reverse the columns and rows in the matrix as

$$S^T = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1m} \\ s_{21} & s_{22} & \dots & s_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{r1} & s_{r2} & \dots & s_{rm} \end{bmatrix};$$

thus, by the product of S and S^T, we obtain two different matrices as

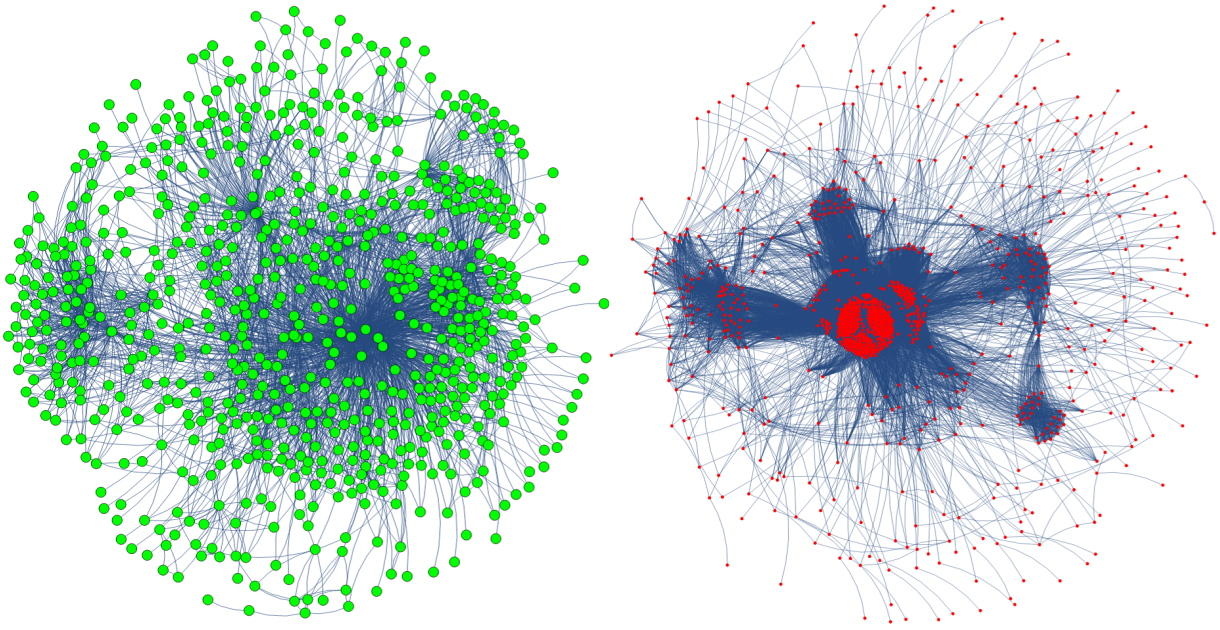
$$S.S^T = \begin{bmatrix} s'_{11} & s'_{12} & \dots & s'_{1m} \\ s'_{21} & s'_{22} & \dots & s'_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s'_{m1} & s'_{m2} & \dots & s'_{mm} \end{bmatrix} \quad \text{and} \quad S^T.S = \begin{bmatrix} s''_{11} & s''_{12} & \dots & s''_{1r} \\ s''_{21} & s''_{22} & \dots & s''_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ s''_{r1} & s''_{r2} & \dots & s''_{rr} \end{bmatrix},$$

where $S.S^T$ is a metabolite-centric matrix and $S^T.S$ is a reaction-centric matrix. Considering a normalising step for those matrices as

$$f(x) = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{if } x \neq 0 \end{cases},$$

one can construct adjacency matrices, $A^m_{ij} = f(s^{'}_{ij})$ and $A^r_{ij} = f(s^{''}_{ij})$, to form graphs like that introduced in Fig. 2.1 in the Association Networks subsection.

The graphs in Fig. 2.5 were generated from A^m and A^r using a stoichiometric matrix belonging to homo sapiens metabolism retrieved from BiGG Models Database [25]. In Fig. 2.5a, the graph nodes stand for the metabolites, and graph edges are the reactions. In contrast, in Fig. 2.5b, the roles are reversed so that the graph edges represent the metabolites, and the graph nodes represent the reactions.



(a) Metabolic-centric Network

(b) Reaction-centric Network

Figure 2.5: Network Representations for Homo Sapiens Metabolic Model.

Studying biological metabolic systems and designed models to achieve cellular objectives like cell growth or Adenosine Triphosphate Production (ATP) necessitates various tools to be integrated with reconstructed GCN's [26, 23]. One of the commonly used tools is Flux Balance Analysis (FBA) as an optimisation scheme. It is a constraint-based modelling approach to simulate microbial metabolisms and can be applied to biochemical-reaction networks containing the chemical transformations and flux exchanges [27, 28].

While one can express the metabolic fluxes in a one-dimensional array (the

so-called flux vector V) as

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_r \end{bmatrix} = (v_i) \in \mathbb{R}. \tag{2.5}$$

V contains flux exchange values for the corresponding reactions in the system and gives information about the flux distribution; hence, those can be both positive and negative real numbers. Defining a mass-balance ($S.V = 0$) constraint in the FBA enables us to analyze the metabolic network operations in a steady-state solution space [27, 28].

$$S.V = \begin{bmatrix} s_{11}v_1 + s_{12}v_2 + \cdots + s_{1r}v_r \\ s_{21}v_1 + s_{22}v_2 + \cdots + s_{2r}v_r \\ \vdots \\ s_{m1}v_1 + s_{m2}v_2 + \cdots + s_{mr}v_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{2.6}$$

The higher amount of metabolite consideration in the set of rules, S, in other words, the larger matrix size by its rows amount means the more complex type of organisation structure taken into account while preserving the steady-state in the whole system.

More than one steady-state solution might be present since it is impossible to identify all constraints in a cellular system [27]. Therefore, one can formulate an optimisation approach to identify reaction network steady-states that maximise the biomass [27, 28] or control the production of specific metabolites [29] within a defined objective function under the consideration of the system constraints. According to Price et al. (2004), there are three primary purposes to generate objective functions [28]:

i. to discover allowable characteristic properties in the genome-scale network reconstruction,

ii. to mimic probable physiological functions like biomass or ATP to be able to determine likely physiological states and

iii. to design a genetic variant or sub-type to obtain a desired particular product.

One can express objective function coefficients in a one-dimensional array as

$$O = \begin{bmatrix} o_1 & o_2 & \dots & o_r \end{bmatrix} = (o_i) \in \mathbb{R}. \tag{2.7}$$

As given in Eq. (2.8), the biomass formulation delivers the output with its non-zero coefficients, which are the decisive ones for the flux elements of V to be considered.

$$O.V = (o_1 v_1 + o_2 v_2 + \cdots + o_r v_r) \in \mathbb{R}_{\geq 0}. \tag{2.8}$$

Stoichiometric (or mass-balance) constraints were introduced so far in Eq. (2.4) and Eq. (2.6). In addition, upper and lower bounds are presented for particular fluxes in V during the optimisation process. The bounds are used in the reactions for uptake and secretion of any organic metabolite. In the uptake reactions, nutrients are transported to the inside of the metabolic network. In the secretion reactions, products are exported to the outside of the network. The rest of the fluxes in V are used in the exchange reactions, namely the intermediate reactions in the network. The constraints influence the reactions for uptake and secretion, whereas no limitation is considered in the exchange reactions. Quantifying imported nutrients and exported outputs (resources and products) by constraining them with upper and lower bounds to fulfil a single objective function goal might significantly influence the optimisation process.

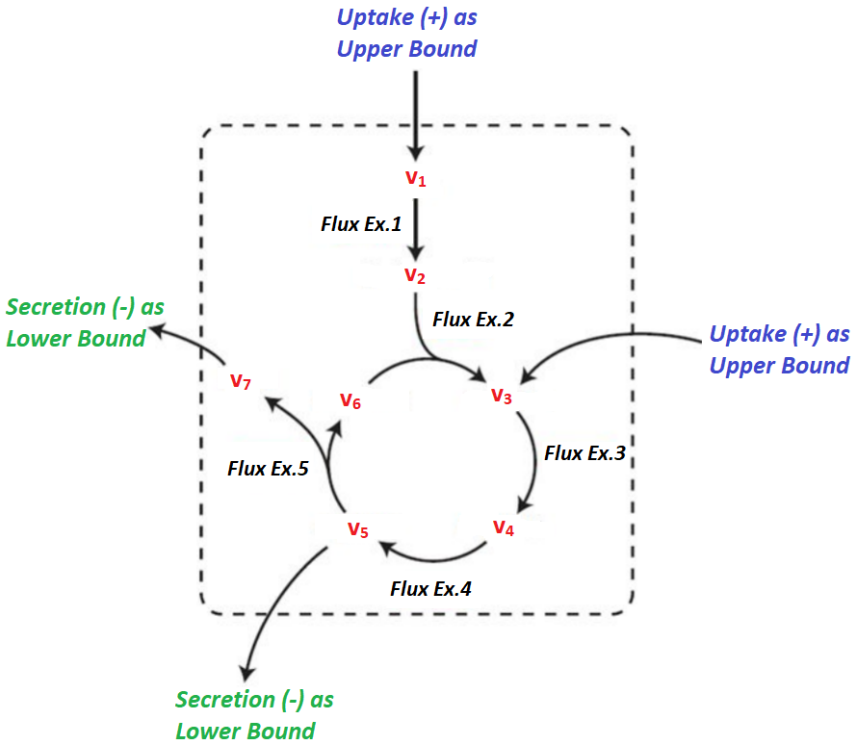


Figure 2.6: A Simplified Reaction-centric Network Sketch Shows the Reactions for Exchange, Uptake and Secretion.

As a summary of the above-explained series of constraints, mass-balance equality (Eq. (2.6)), upper & lower bounds for fluxes (Fig. 2.6), and the objective function (Eq. (2.7)) are the three fundamental constraints that set off a linear programming problem because it is possible to formulate them linearly [28]. The optimisation result: flux vector V (Eq. (2.5)) maximises the objective function in the form of a flux distribution [27, 28]. Since each term in Eq. (2.8) is a produced biomass expression for the fluxes, the summation of those terms will give the overall growth of the system for a single network state.

Different solution vectors of V can be obtained from the linear optimisation process by varying the constraints introduced above. As a compact set of rules, the stoichiometric constraints significantly influence the mass-balance equation; consequently, the solution vector V [30]. A stoichiometric matrix from scratch can be formulated, ensuring the mass-balance constraints are incorporated in the reaction cycles of the investigated system. However, the homo sapiens metabolic model was taken as the set of rules in this thesis work. Varying upper & lower flux bounds and the objective function are the two alternative approaches introduced in the following subsections to understand the model behaviour while the optimisation is carried on.

2.2.1 Resource Utilisation

Environmental conditions such as resource availability affect the pattern of outputs in a metabolic network. In case of fewer resources (nutrients) availability, the active production network gets more interconnected through more flux exchanges to produce the necessary input for the ongoing metabolic reactions. [28, 31, 32, 33]

$$V^b = \{v_1^b, v_2^b, \ldots, v_x^b\} = (-a \leq v_i^b \leq a) \in V \tag{2.9}$$

Let V^b is a list of fluxes with x elements randomly picked from V (Eq. (2.5)) to be limited with the bounds: $(-a, a)$. The same tolerance in both negative and positive direction for the bounds allows the network to treat the respective flux flow as uptake or secretion based on the system need. The fluxes that are not included in V^b are matched with extreme high boundary values so that they are not constrained while the linear optimisation.

$$V^e = \{v_1^e, v_2^e, \ldots, v_y^e\} = (0 \leq v_i^e \leq 0) \in V \tag{2.10}$$

Assigning zero to the upper & lower bounds suppresses the respective flux exchange in the active production network. Those fluxes can not be used for

the uptake, secretion, nor intermediate reactions. V^e (Eq. (2.10)) is the list of fluxes with y elements randomly selected from V (Eq. (2.5)) to be deleted from the network by assigning zero to the bounds.

Such limitations on resources serve as capacity constraints defining the active reactions and reversibility of flux exchanges [30]. Varying x and a in Eq. (2.10), and y in Eq. (2.10), we obtain various biomass values by the linear programming algorithm to fulfil a fixed objective function.

2.2.2 Product Portfolio Diversification

The objective function can be assumed as a production plan that rules the diversity of products that metabolism takes into account to maximise cellular growth [30]. As previously mentioned, this is because the pattern of output biomass (Eq. (2.8)) is governed by the objective function (Eq. (2.7)). Its non-zero coefficients force the network for an optimal solution with their value range and positive and negative signs.

Defining various arrays of objective function coefficients, consisting of elements with negative or positive signs, will allow us to create a diverse group of products that the network is capable of producing. In the same direction, adjusting the number of objective function terms is the second step of that diversification.

Both Resource Utilisation (RU) and Product Portfolio Diversification (PPD) approaches will be discussed in more detail in Chapter 3.

2.3 Integration of Concepts

As explained in the previous section, we obtain maximised flux distribution of the metabolic organism for a single network state by performing linear optimisation. This section clarifies how the outputs from linear optimisation in more than one run are converted into a compatible format to construct data structures similar to steel production data. With this integration step, one can construct association networks derived from FSS and FBS labels of the generated data and calculate modularity values concerning alternative null models.

Fig. 2.7 illustrates the complete simulation model. FBA optimisation scheme is summarised in Fig. 2.7a with a defined set of rules and system constraints, as introduced in the previous section. The overall growth of the biomass (Eq. (2.8)) is obtained with a single run of the optimisation algorithm. The resultant value is a simulation event conceptually equivalent to a steel production event. The optimisation algorithm is run 10000 times to create a data set with 10000 events.

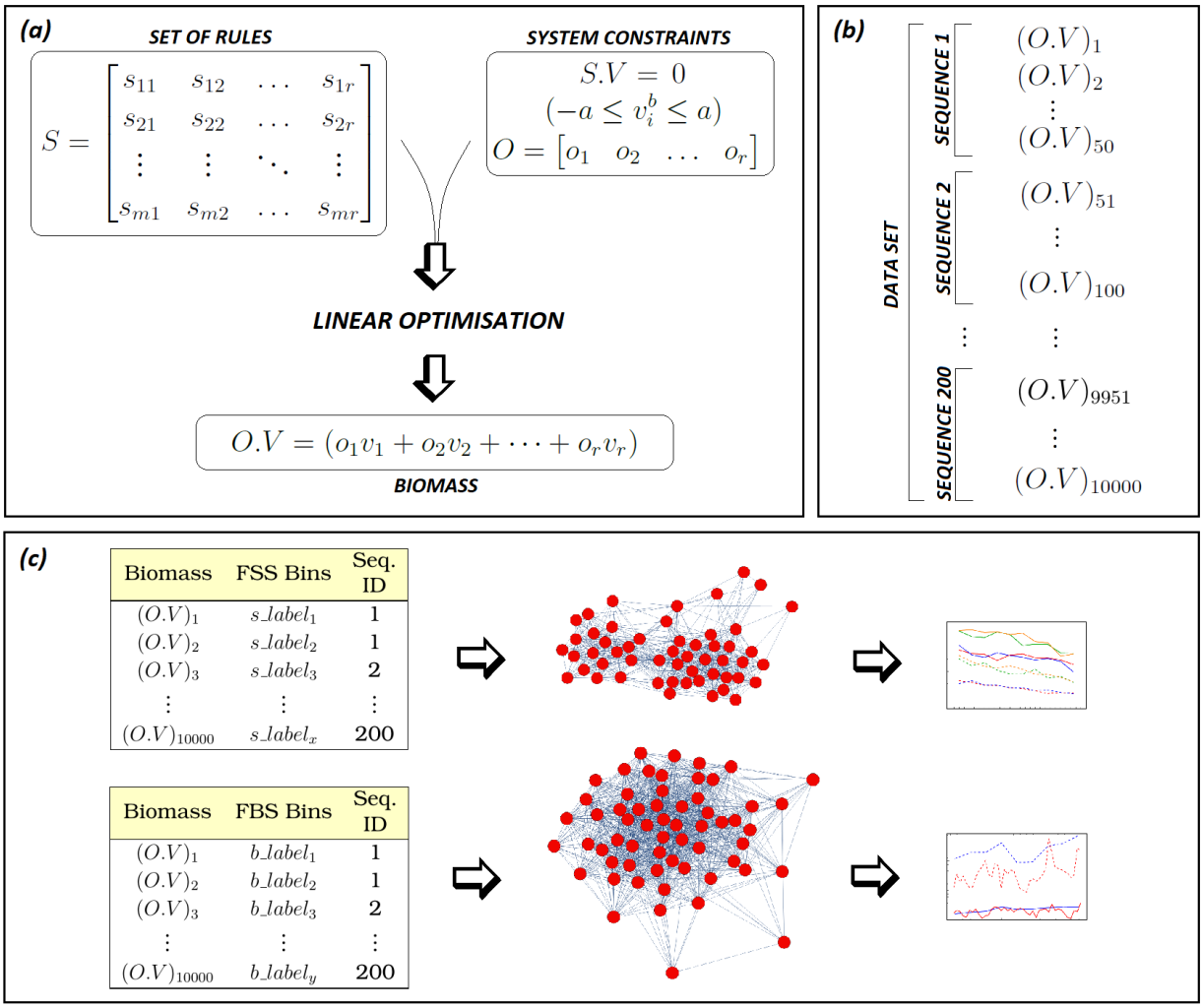


Figure 2.7: Simulation Model Illustration.

Fig. 2.7b shows the data structure generation by introducing a production sequence concept. In steel manufacturing, each production sequence shows consistency among its events based on the production constraints as introduced in Section 1.1; therefore, the random choice for non-zero coefficients in the objective function is kept fixed only for the events in the same sequence. Hence, the optimisation scheme can use various flux series to be considered in the biomass for created events in different sequences. Fig. 2.7c shows the generated data labelled in alternative ways, as introduced in Tables 2.2 and 2.3, which is convenient to construct graphs to be analysed.

3 Implementation, Analysis and Results

In the following two sections, we implement the methods and share the analysis results for the steel production and simulation events. Creating sets of experimental data from the simulation allows comparing the statistical characteristics of their association network with those formed from the production data sets from steel manufacturing.

3.1 Steel Production Events

3.1.1 Data Collection and Cleaning

We started to cleaning process by converting string-type data into floating-point numbers, modifying inconsistent punctuation marks between digits into one typical style and converting null values into the integer value 0. After going over minor revision steps, we introduce some preconditions below to disregard data errors and fill the gaps in the data sets.

- The steel material density was calculated for every event, and a range was taken into account between $6.5x10^{-6}\ kg/mm^3$ and $8.5x10^{-6}\ kg/mm^3$. The production orders with density values out of that range were discarded from consideration.
- The input capacity of machines was identified for width and weight as in the range of 800–2000 mm, and 2669–26690 kg, which allowed to perform necessary unit conversions correctly.
- The unit of length values was considered as millimetre (mm). In case of need, length values were converted into suitable magnitudes based on the above given two preconditions.

Zeros were replaced with calculated values for every event with a maximum of one unknown value considering the preconditions mentioned above and the $density = mass/volume$ equation. The events with two zero values were compared with consecutive events, and missing terms were filled based on the con-

sistency among the same attribute values. Sequences with less than 50 events were removed from the data sets considering those short sequences might be generated for some test processes.

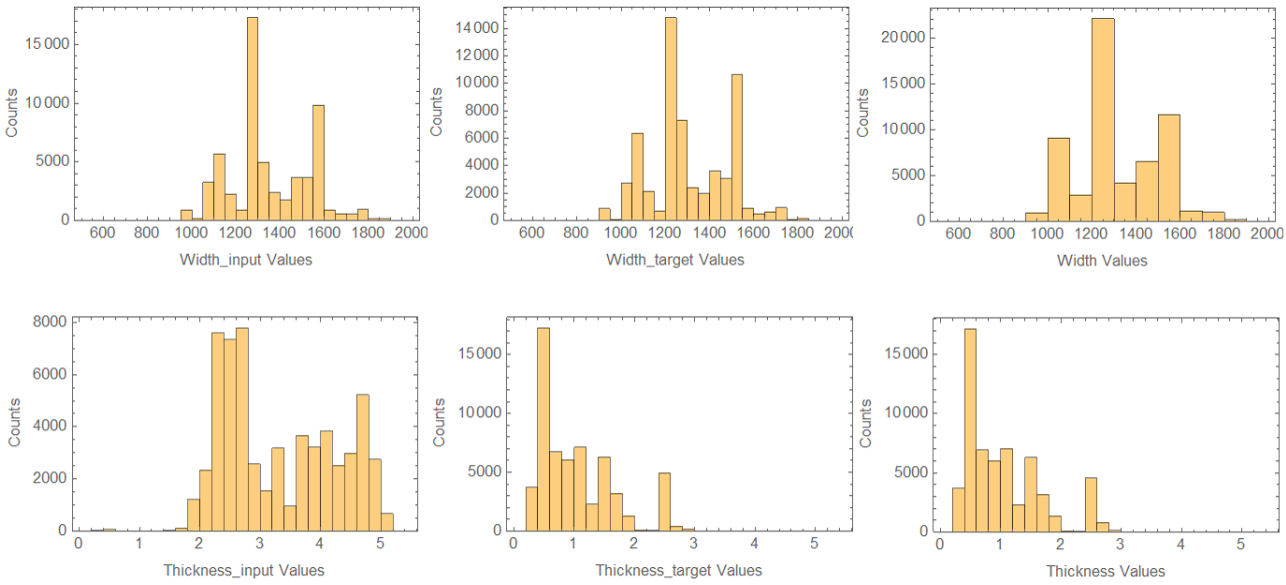


Figure 3.1: PLTCM Data Set Attributes.

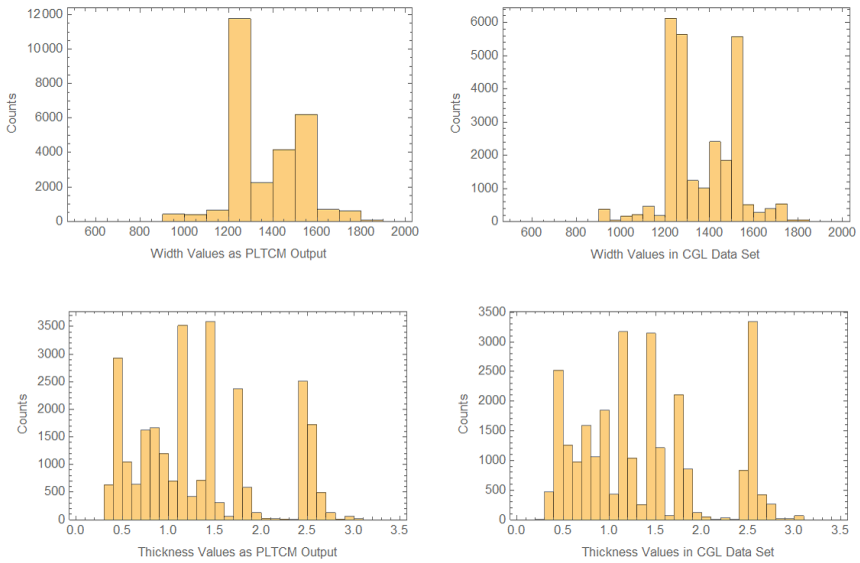


Figure 3.2: Attribute Values for Common Events in PLTCM and CGL Data Sets.

The product features differ before and after the handling by production lines. We are interested in having the feature values before the events are treated in machines to capture the effective patterns in sequence planning concerning our hypothetical constraints. In the PLTCM data collection, width, thickness and

length features have additional attributes as labelled with $input$ and $target$ for each, and their histograms are given in Fig. 3.1. We considered $input$ labelled attributes for PLTCM in our calculations. Unlike PLTCM, the CGL data set has single attributes for dimension features, which we assume as the output values. Based on the reason explained above, we matched the events in the CGL data set with the ones given in PLTCM data set by the attributes, $Material\_ID$ and $Piece\_ID$. The first column in Fig. 3.2 shows the resulting PLTCM data set thickness and width values that we used as $input$ values for CGL.

At the final stage, obtained data set lengths are given below.

- CCM data set: $347,418$ events.
- CSP data set: $205,496$ events.
- PLTCM data set: $59,604$ events.
- CGL data set: $27,147$ events.

3.1.2 Analysis Steps and Results

The complete analysis of steel production events comprises a check for meaningful modularity structures in various dimensions for the association networks generated from the data collection. We introduce those dimensions as;

1. production line,
2. production feature,
3. production constraint,
4. null model,
5. time resolution, and
6. network resolution.

For the first dimension, distinguished data sets belonging to CCM, CSP, PLTCM, and CGL were considered. The following dimensions were examined independently in each production line.

Since width and thickness are the products' physical quantities that are deliberately reformed during the whole manufacturing process, those data feature columns are taken into account for each production line to be analysed as the second dimension.

As we argued previously, two fundamentally different constraints, technology-driven constraints and load-driven constraints, act on the manufacturing process. Alternative binning methods and different network approaches were iden-

tified for those constraints. FSS and FBS networks were generated in the third dimension for each production feature in every production line.

NM-d and NM-m, as previously introduced, were considered to check the randomness of the association networks. As the fourth dimension, those alternative null models were constructed for each FSS and FBS network generated. The resulting z-scores are more straightforward than the resulting modularity values since they take out any effect from different link densities. For this reason, we shared only z-scores as bar charts in this subsection and attached the bar charts and curve plots for the modularity values as supplementary materials.

Time resolution is the fifth dimension, and it consists of different observation-window categories as discrete-time windows, sliding-time windows and complete data with two halves. Each resolution is a means of partitioning the data of historically ordered production events into equal sizes differently. For each window in three categories, the first four dimensions were performed. The analysis with time-resolved fashion allows checking if any significant constraint impact occurs systematically through the time windows created with varying sizes. Since the most significant results were obtained from the last category, the complete data with two halves, we shared and discussed that category in this subsection. The analysis results for discrete-time windows and sliding-time windows are supplementary materials: S4, S5 and S6, S7, S8, S9.

As the last dimension, association networks were obtained from the first four dimensions in the observation-window categories. The sliding-time windows and the complete data with two halves were diversified in two different network resolutions by changing the node number. We achieved this by choosing the appropriate step and bucket sizes while generating graphs. We aimed to obtain the maximum number of nodes to quantify the modularity and keep the node numbers the same in different network approaches in the respective time window. Other than the CSP Thickness and CCM Thickness networks, including fewer nodes than 15 in some cases, all networks have varying node numbers between 25–90. The resulted plots for four production lines in alternative network resolutions are presented in supplementary materials: S6, S7, S8, S9 and S10, S11, S12, S13.

We pretend that the real network structure is more homogeneous than it is and pretend that it is statistically reliable. However, those assumptions are slightly wrong, and those can sometimes lead to cases looking statistically significant even though they are not. An error bar was included in each z-score bar to distinguish the reliably distributed network from the non-robust ones. The error bar is the mean value of the respective z-score by removing and putting back 10% of the data ten times to check the robustness of the statistical signals. The T-shaped symbol represents the standard deviation of the error bars.

Condensed analysis results as bar charts are given in Fig. 3.3, showing z-scores concerning alternative null models in different network approaches for width and thickness features of four production lines. In the bar charts, z-scores are indicated with a colourless line finish, and error bars were included in colour. Green lines are indicated on the values: $+1$ and -1 as the significance thresholds for signals as one standard deviation range.

At first glance on the bar charts, time does not influence modularity change except a few minor differences between the first halves and second halves of the data sets; the complete data column provides the overview of both halves. Moreover, CCM–CSP and PLTCM–CGL pairs have significantly similar results in terms of z-scores for different null models. That case is not unexpected for the first pair since CCM is an inner CSP module; hence, the same production sequence schedule is performed on both production lines.

FSS networks for the width feature in CCM and CSP are always modular and hierarchically organised mainly. On the other hand, FBS networks of the same feature in the same production lines have nonrandom and non-modular hierarchical organisation after checking the robustness. Until perturbing the data, the networks are modular and hierarchical, showing the unstable case of the FBS networks for the width feature in CCM and CSP. FSS networks are modular and organised simple, while FBS networks are non-modular and hierarchically organised regarding the thickness feature in the same production lines.

For the thickness feature in PLTCM and CGL, FSS and FBS networks are very modular. In CGL, they do not comprise any complex textures, and they are robust, whereas they have a hierarchical organisation in PLTCM. For width feature in PLTCM, FSS and FBS networks are again modular and have non-robust hierarchical organisation due to the error bars with long T-shaped symbols. For the width feature in CGL, FSS networks are not modular and hierarchical, while FBS networks have non-robust modular and hierarchical organisation.

Fig. 3.3 shows us that alternative network approaches: FSS and FBS, concerning technology-driven constraints and load-driven constraints, give different results. The difference is rather significant for the width–thickness features in CCM–CSP and the width feature in CGL, whereas it is not very significant for the width–thickness features in PLTCM and the thickness feature in CGL.

In our opinion, FBS networks would have significantly differed from FSS networks for the length and weight features in PLTCM and CGL since those features have high diversity on the production events. We also argue that they would significantly impact load constraints in PLTCM and CGL compared to thickness and width features based on alternative handling categories introduced in Fig. 1.3.

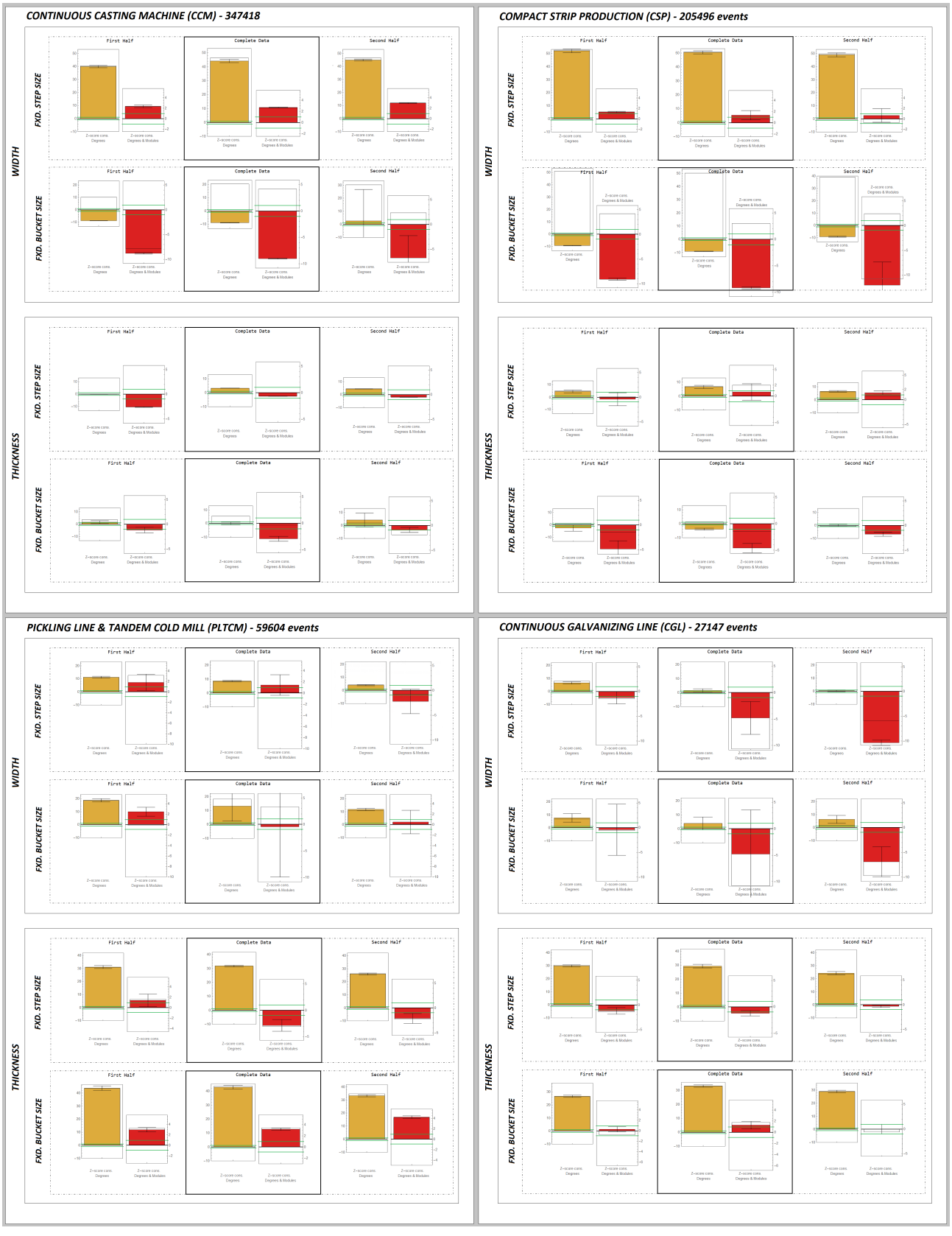


Figure 3.3: CCM, CSP, PLTCM, CGL Analysis Bar Chart Results: Z-scores.

3.2 Simulation Events

3.2.1 Design Steps for the Simulation Model

As summarised in Fig. 2.7, the simulation model performs a single run optimisation with a defined set of rules and system constraints in the first step. As a second step, it generates the data set based on the introduced production sequence concept. Finally, the outcome data set is labelled in alternative binning schemes to construct networks as ready for the network metrics analysis. In this subsection, we quantify the characteristics of the model and vary the identified system constraints used in the second step.

Homo sapiens metabolic model with 738 metabolites and 1008 reactions was used as the set of rules (Eq. 2.4) and kept fixed in our simulation model. The linear optimisation algorithm was run 10000 times to create a data set with 200 sequences, each with 50 events.

A constrained flux list, V^b (Eq. 2.9), was created to be considered in each optimisation run. The fluxes used in the intermediate reactions were given the range of bounds as $(-500, 500)$ since it is impossible to define infinity values in the optimisation algorithm. The bound variable, a, was assigned to 5 for randomly chosen 105 fluxes out of 1008 as the fluxes used in uptake and secretion reactions.

Regarding the previously introduced production sequence concept, biomass (Eq. 2.8) was designed slightly sparse, consists of randomly selected fluxes matched with non-zero coefficients dedicated to each sequence instead of all 1008 fluxes. The counts of biomass series for 200 sequences is attached as supplementary material: S14.

We recall the Product Portfolio Diversification (PPD) and Resource Utilisation (RU) introduced in Section 2.2 to vary the system constraints and derive design steps as the basis of the simulation events analysis;

i. PPD-1 picks random floating-point numbers from four different intervals; $(-1, 1)$, $(-4, 4)$, $(-4, -2)$, and $(2, 4)$ as the objective function coefficients (Eq. 2.7) which results as four distinctive data sets after the optimisation runs completed,

ii. RU-1 generates distinguished deletion lists (V^e, Eq. 2.10), each consists of randomly picked fluxes with the numbers range from 50 to 450 in step 50 and considers those in the four distinctive data sets mentioned above to generate nine different variations of each,

iii. RU-2 updates 105 to 420, which was previously defined as the number of randomly chosen fluxes in V^b to be limited with bound value, a, results

with one more variation of the above steps, and

iv. PPD-2 creates three revised versions of the designed biomass series by randomly choosing the reduced number of terms by 25%, 50%, and 75%, respectively, resulting in three more variations of the above design steps.

In short, RU-1 and RU-2 manipulate the simulation model's resource usage by knocking some reactions out and limiting fluxes used for uptake & secretion reactions. PPD-1 enforces the production plans based on their directionality by imposing symmetric and nonsymmetric number intervals. PPD-2 provides varieties between a vast production range capacity and a more specific one by adjusting the richness of objective functions.

3.2.2 Analysis Results

Derived data sets from the design steps introduced in the above subsection were considered to construct networks concerning alternative binning methods, calculate modularity value of the networks and calculate z-scores concerning two alternative null models as performed in steel production events analysis.

The resulted curve plots are presented as supplementary materials: S15, S16, S17, and S18. As the deleted reaction number increases, modularity shifts down together with a significant decrease in NM-d z-scores. In contrast, the NM-m z-scores are always around zero. A condensed version of those resulted curve plots was re-arranged by taking the mean of both modularity values and z-scores in each data set variation derived from the RU-2 and PPD-2 steps. The results are given in Fig. 3.4.

None of the curves in FSS networks has a change in modularity. In contrast, FBS networks modularity increases for only green and orange coloured curves. From left to right, we increase the constraint that the production plan (or product portfolio) impose on the whole production process. The product portfolio is grouped into speciality products on the right end. The effect of the portfolio changing occurs for the production plans enforced in one direction and does not occur for the symmetrically organised production plans. The red and blue curves are less attractive and instead serve as an orientation since their objective function coefficients are symmetric around zero and damp each other, resulting in a not occurring product. Therefore, we do not expect them to be severely affected by changes in the richness of the objective function, and that is indeed what we numerically observed.

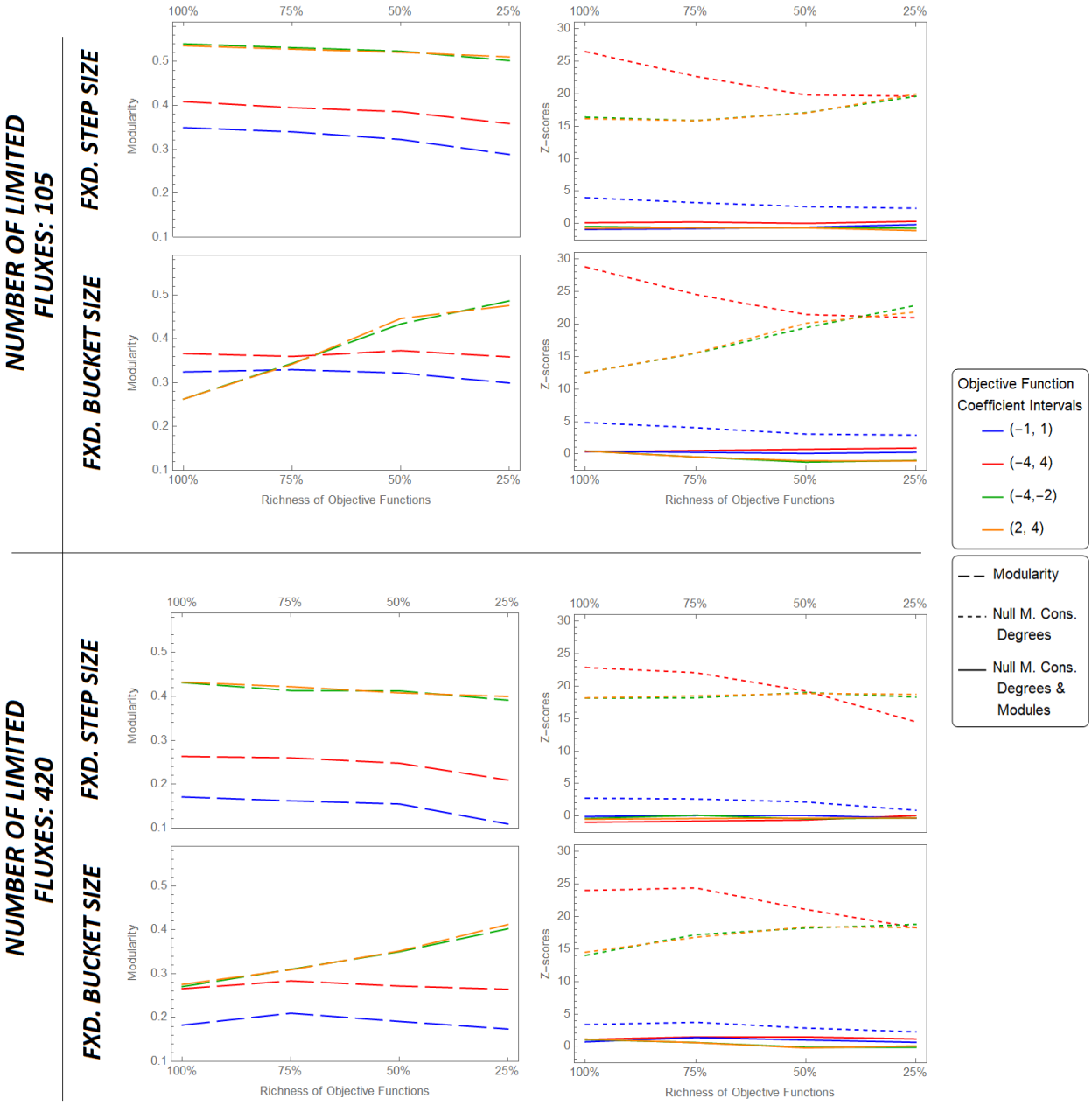


Figure 3.4: Condensed Curve Plots for Simulation Analysis: the Relationship Between Modularity and Richness of Objective Functions.

4 Conclusion and Outlook

We linked these two data processing schemes: Fixed Step Sized binning and Fixed Bucket Sized binning, to different constraint categories and obtained different results in the steel production events analysis. Those results showed us how the association networks' topological features could hint at the constraints of a steel production system. However, the study of links between intrinsic constraints in a production system and some patterns we can observe in the production data is much larger than what started as an investigation in this thesis work.

The optimisation scheme integrated theoretical framework let us conduct simulation experiments. With Flux Balance Analysis, we constrained the system at different levels and allowed for fluctuations in the input and controlled the products via objective functions.

Our abstract theoretical framework is a convenient starting point to conduct perturbation experiments for the existing production networks. On a further step, it should be structured with a random graph considering some additional consistency constraints instead of the one we picked from homo sapiens metabolism in this thesis work. In that random graph, one needs to make sure that the cycles in the graph are suitable to create reactions out of nothing, and some mass balance constraints need to be incorporated.

The main challenge would be to construct arbitrary networks within the Operations Research framework. Therefore, different categories of constraints such as technical constraints, logistic constraints, physical and chemical constraints, economical constraints, and performance-indicator based constraints need to be quantified carefully in this framework.

5 Bibliography

[1] H. Bhadeshia and R. Honeycombe, *Steels: microstructure and properties*. Butterworth-Heinemann, 2017.

[2] C. Tasan, M. Diehl, D. Yan, M. Bechtold, F. Roters, L. Schemmann, C. Zheng, N. Peranio, D. Ponge, M. Koyama, K. Tsuzaki, and D. Raabe, "An overview of dual-phase steels: Advances in microstructure-oriented processing and micromechanically guided design," *Annual Review of Materials Research*, vol. 45, no. 1, pp. 391–431, 2015.

[3] T. Sinha-Spinks, "An infographic of the iron and steel manufacturing process." https://www.thermofisher.com/content/dam/tfs/ATG/CAD/CAD%20Documents/Product%20Manuals%20&%20Specifications/Elemental%20Analysis/Thermo-Scientific-Iron-Steel-process-A3.pdf, 2015. Accessed on 29-06-2021.

[4] P. Cowling, "Design and implementation of an effective decision support system: A case study in steel hot rolling mill scheduling," *Human performance in planning and scheduling*, pp. 217–230, 2001.

[5] A. Özgür, Y. Uygun, and M.-T. Hütt, "A review of planning and scheduling methods for hot rolling mills in steel production," *Computers & Industrial Engineering*, vol. 151, p. 106606, 2021.

[6] SMS group, "All plants from one source." https://www.sms-group.com/press-media/media/downloads/download-detail/download/15940. Accessed on 05-07-2021.

[7] S. Saadaoui, M. Tabaa, F. Monteiro, M. Chehaitly, and A. Dandache, "Discrete wavelet packet transform-based industrial digital wireless communication systems," *Information*, vol. 10, p. 104, 03 2019.

[8] Y. Takase and W. Li, "Strength analysis for shrink fitting system used for ceramics rolls in the continuous pickling line," 2011.

[9] D. Merten, M.-T. Hütt, and Y. Uygun, "A network analysis of decision strategies of human experts in steel production," *submitted to IISE Transactions*, 2020.

[10] A.-L. Barabási, *Network Science*. Cambridge University Press, 2016.

[11] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[12] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[13] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[14] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, p. 026113, Feb 2004.

[15] S. Maslov and K. Sneppen, "Specificity and stability in topology of protein networks," *Science*, vol. 296, no. 5569, pp. 910–913, 2002.

[16] M. Enders, M.-T. Hütt, and J. M. Jeschke, "Drawing a map of invasion biology based on a network of hypotheses," *Ecosphere*, vol. 9, no. 3, p. e02146, 2018.

[17] C. Fretter, M. Müller-Hannemann, and M.-T. Hütt, "Subgraph fluctuations in random graphs," *Phys. Rev. E*, vol. 85, p. 056119, May 2012.

[18] Wikimedia Commons, "The re-drawn chart comparing the various grading methods in a normal distribution, includes standard deviations, cumulative percentages, percentile equivalents, z-scores and t-scores." https://commons.wikimedia.org/wiki/File:The_Normal_Distribution.svg#/media/File:The_Normal_Distribution.svg, 2007. Accessed on 27-06-2021.

[19] A.-L. Barabási, E. Ravasz, and T. Vicsek, "Deterministic scale-free networks," *Physica A: Statistical Mechanics and its Applications*, vol. 299, no. 3, pp. 559–564, 2001.

[20] E. Ravasz and A.-L. Barabási, "Hierarchical organization in complex networks," *Phys. Rev. E*, vol. 67, p. 026112, 02 2003.

[21] M. P. Young, C. Hilgetag, M. A. O'Neill, and M. P. Young, "Hierarchical organization of macaque and cat cortical sensory systems explored with a novel network processor," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 355, no. 1393, pp. 71–89, 2000.

[22] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[23] T. Hao, D. Wu, L. Zhao, Q. Wang, E. Wang, and J. Sun, "The genome-scale

integrated networks in microorganisms," *Frontiers in Microbiology*, vol. 9, p. 296, 2018.

[24] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, *Systems biology in practice: concepts, implementation and application*. John Wiley & Sons, 2005.

[25] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis, "BiGG Models: A platform for integrating, standardizing and sharing genome-scale models," *Nucleic Acids Research*, vol. 44, pp. D515–D522, 10 2015.

[26] B. Kim, W. J. Kim, D. I. Kim, and S. Y. Lee, "Applications of genome-scale metabolic network model in metabolic engineering," *Journal of Industrial Microbiology and Biotechnology*, vol. 42, pp. 339–348, 03 2015.

[27] K. J. Kauffman, P. Prakash, and J. S. Edwards, "Advances in flux balance analysis," *Current Opinion in Biotechnology*, vol. 14, no. 5, pp. 491–496, 2003.

[28] N. D. Price, J. L. Reed, and B. O. Palsson, "Genome-scale models of microbial cells: evaluating the consequences of constraints," *Nature Reviews Microbiology*, vol. 2, no. 11, pp. 886–897, 2004.

[29] A. Varma, B. W. Boesch, and B. O. Palsson, "Biochemical production capabilities of escherichia coli," *Biotechnology and Bioengineering*, vol. 42, no. 1, pp. 59–73, 1993.

[30] J. S. Edwards, R. U. Ibarra, and B. O. Palsson, "In silico predictions of escherichia coli metabolic capabilities are consistent with experimental data," *Nature Biotechnology*, vol. 19, pp. 125–130, 2001.

[31] R. Mahadevan and C. Schilling, "The effects of alternate optimal solutions in constraint-based genome-scale metabolic models," *Metabolic Engineering*, vol. 5, no. 4, pp. 264–276, 2003.

[32] J. L. Reed and B. O. Palsson, "Genome-scale in silico models of e. coli have multiple equivalent phenotypic states: Assessment of correlated reaction subsets that comprise network states," *Genome Research*, vol. 14, no. 9, pp. 1797–1805, 2004.

[33] A. P. Burgard, S. Vaidyaraman, and C. D. Maranas, "Minimal reaction sets for escherichia coli metabolism under different growth requirements and uptake environments," *Biotechnology Progress*, vol. 17, no. 5, pp. 791–797, 2001.

Supplementary Materials

S1: SQL Inquiries for CCM and CSP.

S2: SQL Inquiry for PLTCM.

S3: SQL Inquiry for CGL.

S4: CCM, CSP Analysis Curve Plot Results: Modularity Values and Z-scores in Discrete-time Windows.

S5: PLTCM, CGL Analysis Curve Plot Results: Modularity Values and Z-scores in Discrete-time Windows.

S6: CCM Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

S7: CSP Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

S8: PLTCM Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

S9: CGL Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

S10: CCM Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

S11: CSP Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

S12: PLTCM Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

S13: CGL Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

S14: Counts for Biomass Series.

S15: Simulation Results with Initial Terms in Objective Functions.

S16: Simulation Results with 25% Reduced Terms in Objective Functions.

S17: Simulation Results with 50% Reduced Terms in Objective Functions.

S18: Simulation Results with 75% Reduced Terms in Objective Functions.

```
SELECT ros.r_os_id, ros.production_line_name, ccm.sequence_id, ros.reference_date, NVL(TO_CHAR(slab.piece_id),'NA') piece_id, NVL(TO_CHAR(slab.material_id),'NA') material_id, NVL(TO_CHAR(slab.mold_width),'NA') mold_width, NVL(TO_CHAR(mat.width),'NA') width, NVL(TO_CHAR(mat.thickness),'NA') thickness, NVL(TO_CHAR(mat.weight),'NA') weight, NVL(TO_CHAR(mat.length),'NA') length, NVL(TO_CHAR(mat.heat_id),'NA') heat_id, NVL(TO_CHAR(mat.steel_grade_id_int),'NA') steel_grade_id_int, NVL(TO_CHAR(slab.exit_temp),'NA') exit_temp, NVL(TO_CHAR(mat.slab_transition),'NA') slab_transition
FROM L3MAIN.r_os ros
LEFT JOIN L3MAIN.r_ccm ccm ON ros.r_os_id=ccm.r_os_id
LEFT JOIN L3MAIN.r_ccm_slab slab ON ros.r_os_id=slab.r_os_id
LEFT JOIN L3MAIN.r_mat mat ON ros.r_os_id=mat.r_os_id
WHERE sequence_id IS NOT NULL;
```

```
SELECT DISTINCT ccm2.sequence_id, sl.production_line_name, sl.piece_id, sl.material_id, sl.steel_grade_id_int, sl.heat_id, sl.slab_transition, sl.width, sl.length, sl.weight, sl.thickness, sl.thickness_hsm, sl.cut_time
FROM r_ccm ccm2
LEFT JOIN (
        SELECT ros.production_line_name, ccm.sequence_id, NVL(TO_CHAR(slab.piece_id),'null') piece_id, NVL(TO_CHAR(slab.material_id),'null') material_id, NVL(TO_CHAR(slab.mold_width),'null') mold_width, NVL(TO_CHAR(slab.casting_speed),'null') casting_speed, NVL(TO_CHAR(slab.exit_temp),'null') exit_temp, NVL(TO_CHAR(mat.steel_grade_id_int),'null') steel_grade_id_int, NVL(TO_CHAR(mat.heat_id),'null') heat_id, NVL(TO_CHAR(mat.slab_transition),'null') slab_transition, NVL(TO_CHAR(mat.width),'null') width, NVL(TO_CHAR(mat.length),'null') length, NVL(TO_CHAR(mat.weight),'null') weight, NVL(TO_CHAR(mat.thickness),'null') thickness, NVL(TO_CHAR(mat2.thickness),'null') as thickness_hsm, NVL(TO_CHAR(slab.cut_time),'null') cut_time
        FROM r_os ros, r_ccm_slab slab, r_ccm ccm, r_mat mat, r_mat mat2, r_os ros2
        WHERE mat2.material_id=mat.material_id AND mat2.r_os_id=ros2.r_os_id AND ros2.production_line_name LIKE 'HSM%' AND mat.material_id=slab.material_id AND slab.r_os_id=ccm.r_os_id AND mat.material_type='S' AND mat.modification_date=(
                SELECT MAX(mat2.modification_date)
                FROM r_mat mat2
                WHERE mat2.material_type='S' AND mat2.material_id=slab.material_id
        ) AND ros.r_os_id=mat.r_os_id AND ros.production_line_name LIKE 'CCM1'
) sl ON sl.sequence_id=ccm2.sequence_id
WHERE ccm2.ladle_arrival_time>to_date('01.07.2017', 'DD.MM.YYYY');
```

Figure S1: SQL Inquiries for CCM and CSP.

```
SELECT DISTINCT seq.program_id, seq.program_state, data.piece_id, data.material_id, data.material_sub_type, data.steel_grade_id_int, data.width,
    data.thickness_hsm, data.thickness, data.crosssection, data.weight, data.length, data.pickling_temp_avg, data.pickling_speed_avg, data.
    pickling_pressure_avg, data.elongation, data.oiling_flag, data.oil_type, data.operation_mode, data.roll_set_id, data.spm_mode, data.
    yield_point_calc, data.trim_flag, data.trim_width, data.cut_date, data.target_thickness, data.pl_oiling_flag, data.pl_oiling_type, data.
    pl_oiling_weight_top, data.pl_elongation, data.hot_coiling_temp, data.hrc_tensile_str, data.hrc_yield_point, data.input_thickness, data.
    input_width, data.input_length, data.target_width, data.target_length
FROM pg seq
LEFT JOIN (
        SELECT pgl.program_id, NVL(TO_CHAR(pgl.material_id),'null') material_id, NVL(TO_CHAR(mat.piece_id),'null') piece_id, NVL(TO_CHAR(mat.
    material_sub_type),'null') material_sub_type, NVL(TO_CHAR(mat.steel_grade_id_int),'null') steel_grade_id_int, NVL(TO_CHAR(mat.width),'null')
    width, NVL(TO_CHAR(mat_hot.thickness),'null') thickness_hsm, NVL(TO_CHAR(mat.thickness),'null') thickness, NVL(TO_CHAR(mat.thickness*mat.width
    ),'null') crosssection, NVL(TO_CHAR(mat.weight),'null') weight, NVL(TO_CHAR(mat.length),'null') length, NVL(TO_CHAR(PLTCM.pickling_temp_avg),'
    null') pickling_temp_avg, NVL(TO_CHAR(PLTCM.pickling_speed_avg),'null') pickling_speed_avg, NVL(TO_CHAR(PLTCM.pickling_pressure_avg),'null')
    pickling_pressure_avg, NVL(TO_CHAR(tcm.elongation),'null') elongation, NVL(TO_CHAR(tcm.oiling_flag),'null') oiling_flag, NVL(TO_CHAR(tcm.
    oil_type),'null') oil_type, NVL(TO_CHAR(tcm.operation_mode),'null') operation_mode, NVL(TO_CHAR(tcm.roll_set_id),'null') roll_set_id, NVL(
    TO_CHAR(tcm.spm_mode),'null') spm_mode, NVL(TO_CHAR(tcm.yield_point_calc),'null') yield_point_calc, NVL(TO_CHAR(tcm.trim_flag),'null')
    trim_flag, NVL(TO_CHAR(tcm.trim_width),'null') trim_width, NVL(TO_CHAR(tcm.cut_date),'null') cut_date, NVL(TO_CHAR(pdi.target_thickness),'null
    ') target_thickness, NVL(TO_CHAR(pdi.pl_oiling_flag),'null') pl_oiling_flag, NVL(TO_CHAR(pdi.pl_oiling_type),'null') pl_oiling_type, NVL(
    TO_CHAR(pdi.pl_oiling_weight_top),'null') pl_oiling_weight_top, NVL(TO_CHAR(pdi.pl_elongation),'null') pl_elongation, NVL(TO_CHAR(pdi.
    hot_coiling_temp),'null') hot_coiling_temp, NVL(TO_CHAR(pdi.hrc_tensile_str),'null') hrc_tensile_str, NVL(TO_CHAR(pdi.hrc_yield_point),'null')
     hrc_yield_point, NVL(TO_CHAR(pdi.input_thickness),'null') input_thickness, NVL(TO_CHAR(pdi.input_width),'null') input_width, NVL(TO_CHAR(pdi.
    input_length),'null') input_length, NVL(TO_CHAR(pdi.target_width),'null') target_width, NVL(TO_CHAR(pdi.target_length),'null') target_length
        FROM pdi_pltcm pdi, pgl pgl, r_mat mat, r_mat mat_hot, r_PLTCM_IN PLTCM, r_TCM tcm
        WHERE mat.material_id=tcm.material_id AND mat.material_id=pdi.material_id AND mat_hot.material_id=pgl.material_id AND mat.material_id=pgl.
    material_id AND mat.material_id=PLTCM.material_id AND mat.material_type LIKE 'CC' AND mat.modification_date=(
                SELECT MAX(modification_date)
                FROM r_mat mat2
                WHERE mat2.material_id=mat.material_id AND mat2.material_type='CC'
        ) AND mat_hot.modification_date=(
                SELECT MAX(modification_date)
                FROM r_mat mat3
                WHERE mat3.material_id=mat.material_id AND mat3.material_type='CH'
        )
) data ON data.program_id=seq.program_id
WHERE seq.production_line_name LIKE 'PLTCM%' AND seq.start_actual>to_date('01.01.2018','DD.MM.YYYY');
```

Figure S2: SQL Inquiry for PLTCM.

```
SELECT DISTINCT seq.program_id, seq.program_state, data.material_id, data.piece_id, data.material_sub_type, data.steel_grade_id_int, data.
    spm_elongation, data.temp_end_dff_aim, data.temp_end_rtf_aim, data.temp_end_soak_aim, data.temp_end_slow_cool, data.temp_end_rapid_cool, data.
    coat_wt_top_aim, data.coat_wt_bottom_aim, data.tlv_elongation, data.width, data.thickness, data.crosssection, data.weight, data.length, data.
    galv_top, data.galv_bot, data.elongation_spm, data.roll_set_id, data.elongation_tlv, data.oiling_ind, data.cut_date
FROM pg seq
LEFT JOIN (
        SELECT pgl.program_id, NVL(TO_CHAR(pgl.material_id),'null') material_id, NVL(TO_CHAR(mat.material_sub_type),'null') material_sub_type, NVL
    (TO_CHAR(mat.steel_grade_id_int),'null') steel_grade_id_int, NVL(TO_CHAR(mat.width),'null') width, NVL(TO_CHAR(mat.thickness),'null')
    thickness, NVL(TO_CHAR(mat.thickness*mat.width),'null') crosssection, NVL(TO_CHAR(mat.weight),'null') weight, NVL(TO_CHAR(mat.length),'null')
    length, NVL(TO_CHAR(mat.galv_top),'null') galv_top, NVL(TO_CHAR(mat.galv_bot),'null') galv_bot, NVL(TO_CHAR(cgl.piece_id),'null') piece_id,
    NVL(TO_CHAR(cgl.elongation_spm),'null') elongation_spm, NVL(TO_CHAR(cgl.roll_set_id),'null') roll_set_id, NVL(TO_CHAR(cgl.elongation_tlv),'
    null') elongation_tlv, NVL(TO_CHAR(cgl.oiling_ind),'null') oiling_ind, NVL(TO_CHAR(cgl.cut_date),'null') cut_date, NVL(TO_CHAR(pdi.
    spm_elongation),'null') spm_elongation, NVL(TO_CHAR(pdi.temp_end_dff_aim),'null') temp_end_dff_aim, NVL(TO_CHAR(pdi.temp_end_rtf_aim),'null')
    temp_end_rtf_aim, NVL(TO_CHAR(pdi.temp_end_soak_aim),'null') temp_end_soak_aim, NVL(TO_CHAR(pdi.temp_end_slow_cool),'null') temp_end_slow_cool
    , NVL(TO_CHAR(pdi.temp_end_rapid_cool),'null') temp_end_rapid_cool, NVL(TO_CHAR(pdi.coat_wt_top_aim),'null') coat_wt_top_aim, NVL(TO_CHAR(pdi.
    coat_wt_bottom_aim),'null') coat_wt_bottom_aim, NVL(TO_CHAR(pdi.tlv_elongation),'null') tlv_elongation
        FROM pdi_cgl pdi, pgl pgl, r_mat mat, r_cgl cgl
        WHERE mat.material_id=pdi.material_id AND mat.material_id=pgl.material_id AND mat.material_id=cgl.material_id AND mat.material_type LIKE '
    CG' AND mat.modification_date=(
                SELECT MAX(modification_date)
                FROM r_mat mat2
                WHERE mat2.material_id=mat.material_id
                )
        ) data ON data.program_id=seq.program_id
WHERE seq.production_line_name LIKE 'CGL%' AND seq.start_actual>to_date('01.01.2018','DD.MM.YYYY');
```

Figure S3: SQL Inquiry for CGL.

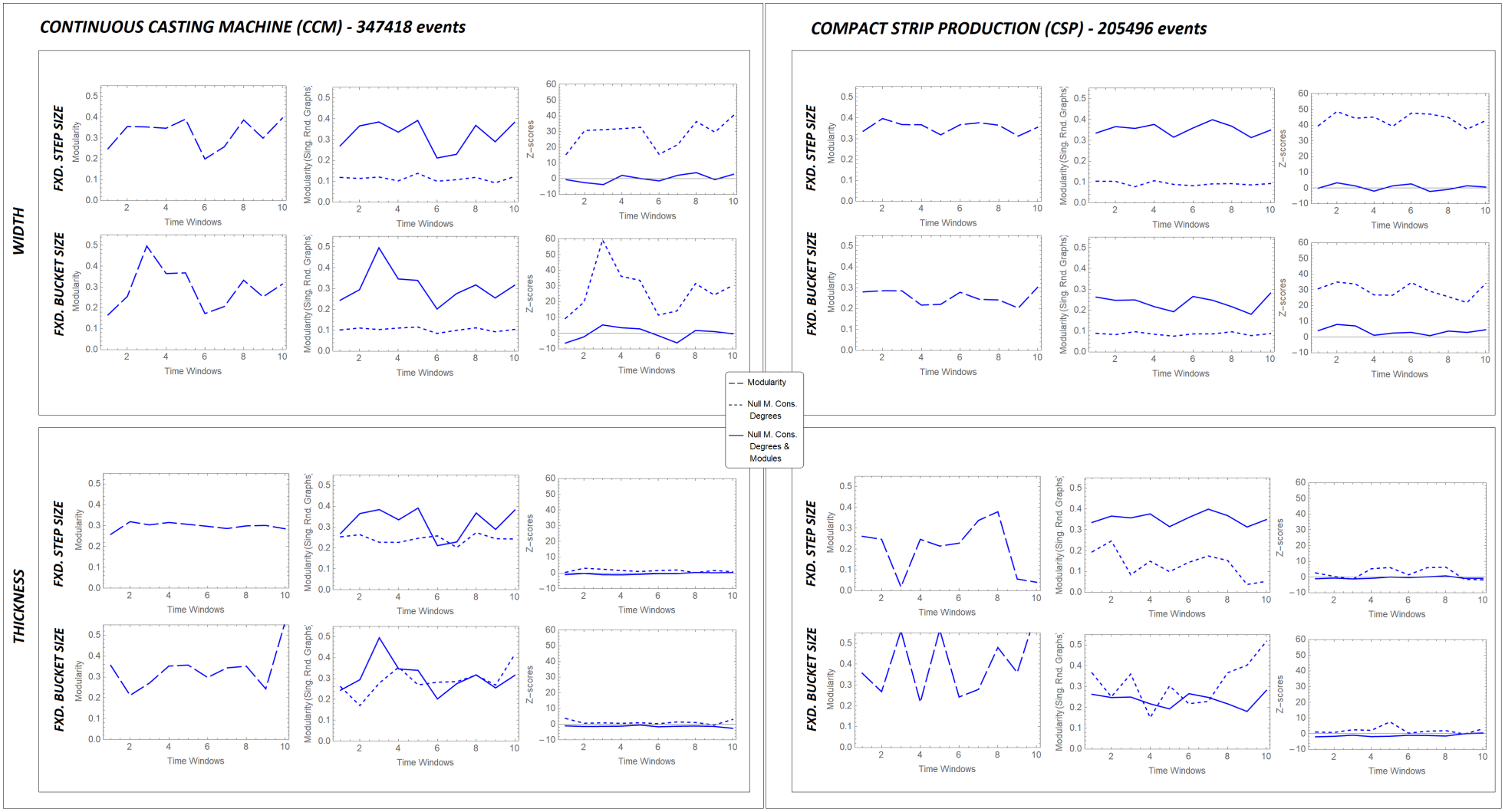


Figure S4: CCM, CSP Analysis Curve Plot Results: Modularity Values and Z-scores in Discrete-time Windows.

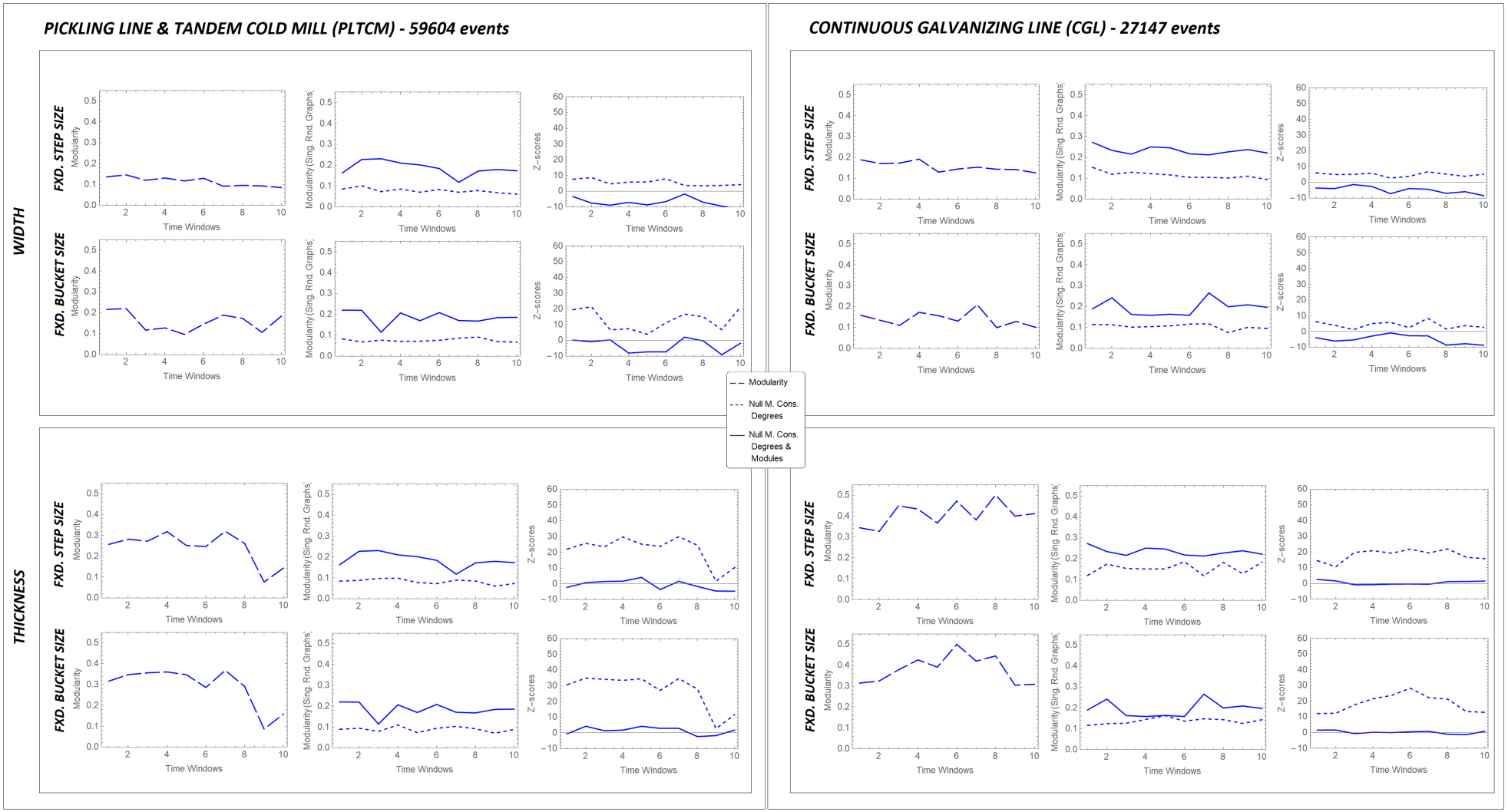


Figure S5: PLTCM, CGL Analysis Curve Plot Results: Modularity Values and Z-scores in Discrete-time Windows.

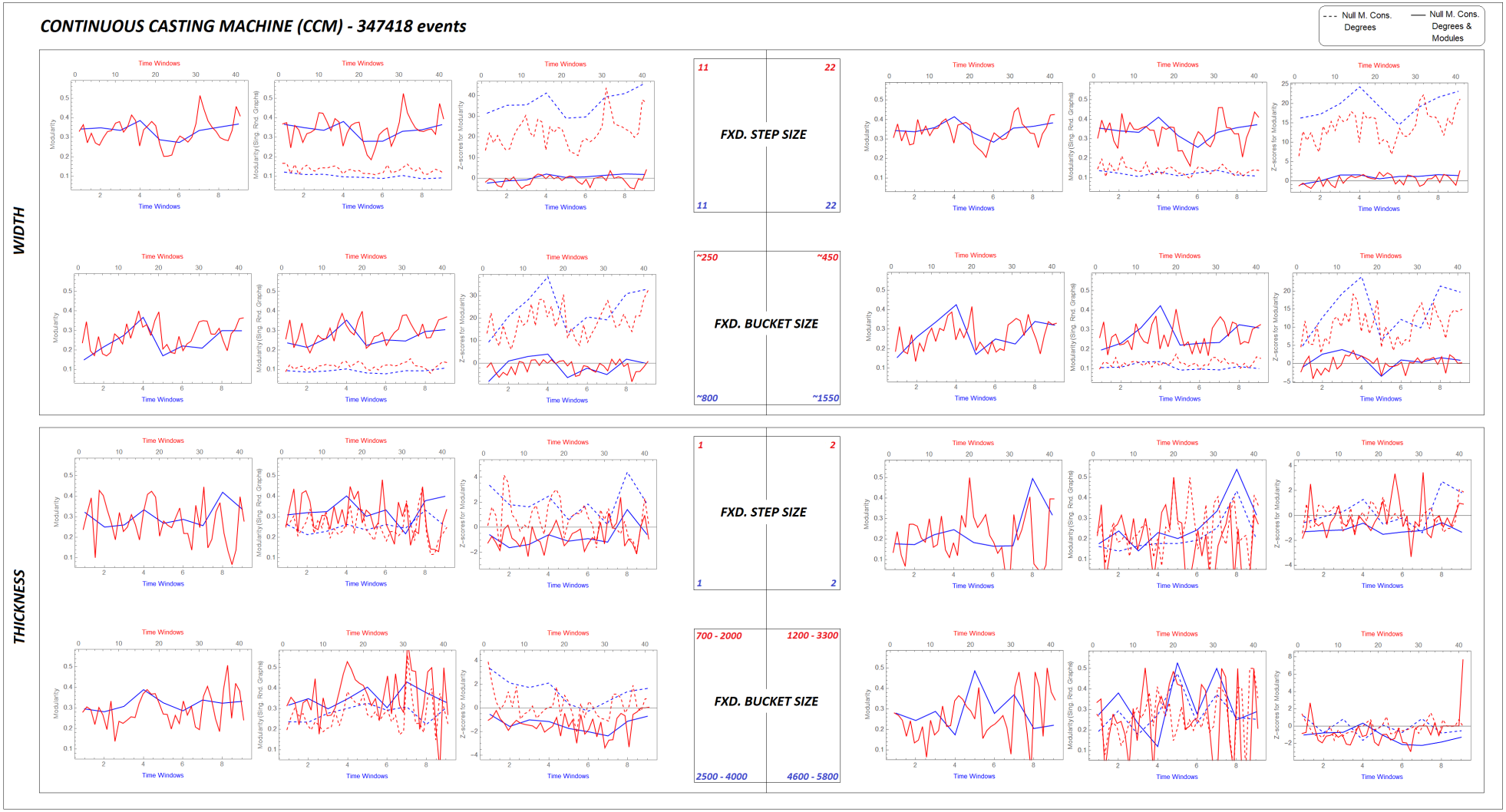


Figure S6: CCM Production Line Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

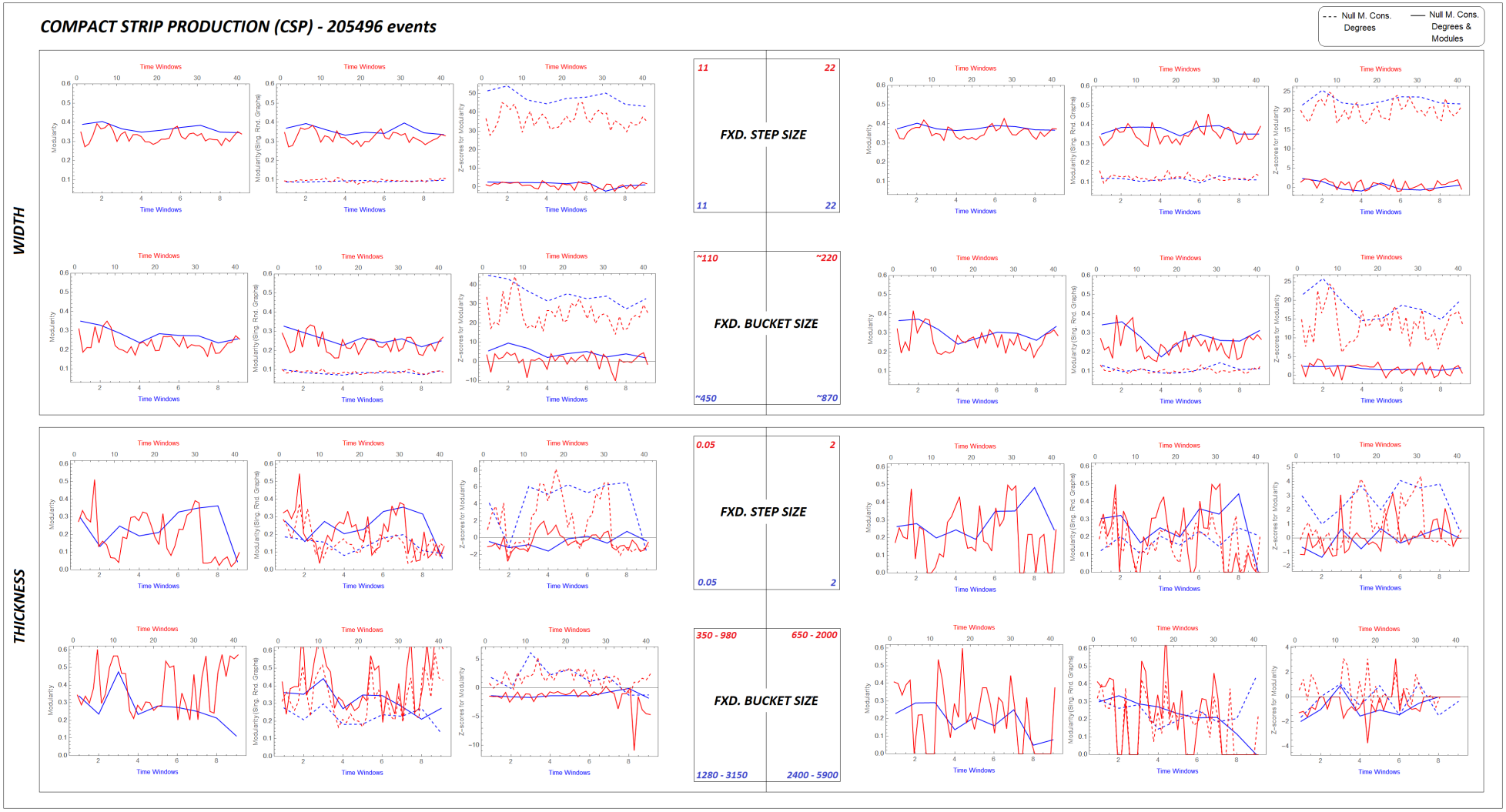


Figure S7: CSP Production Line Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

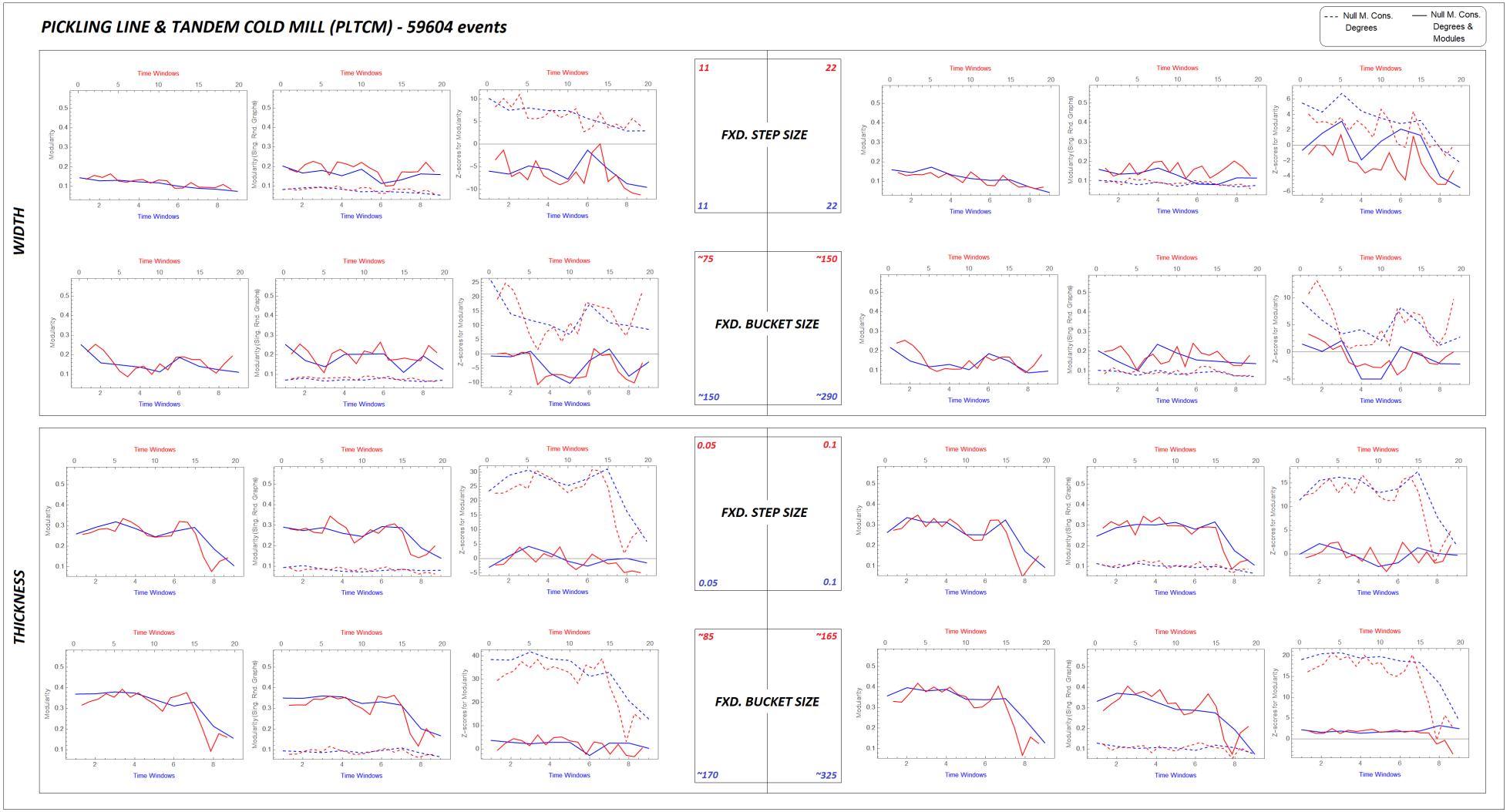


Figure S8: PLTCM Production Line Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

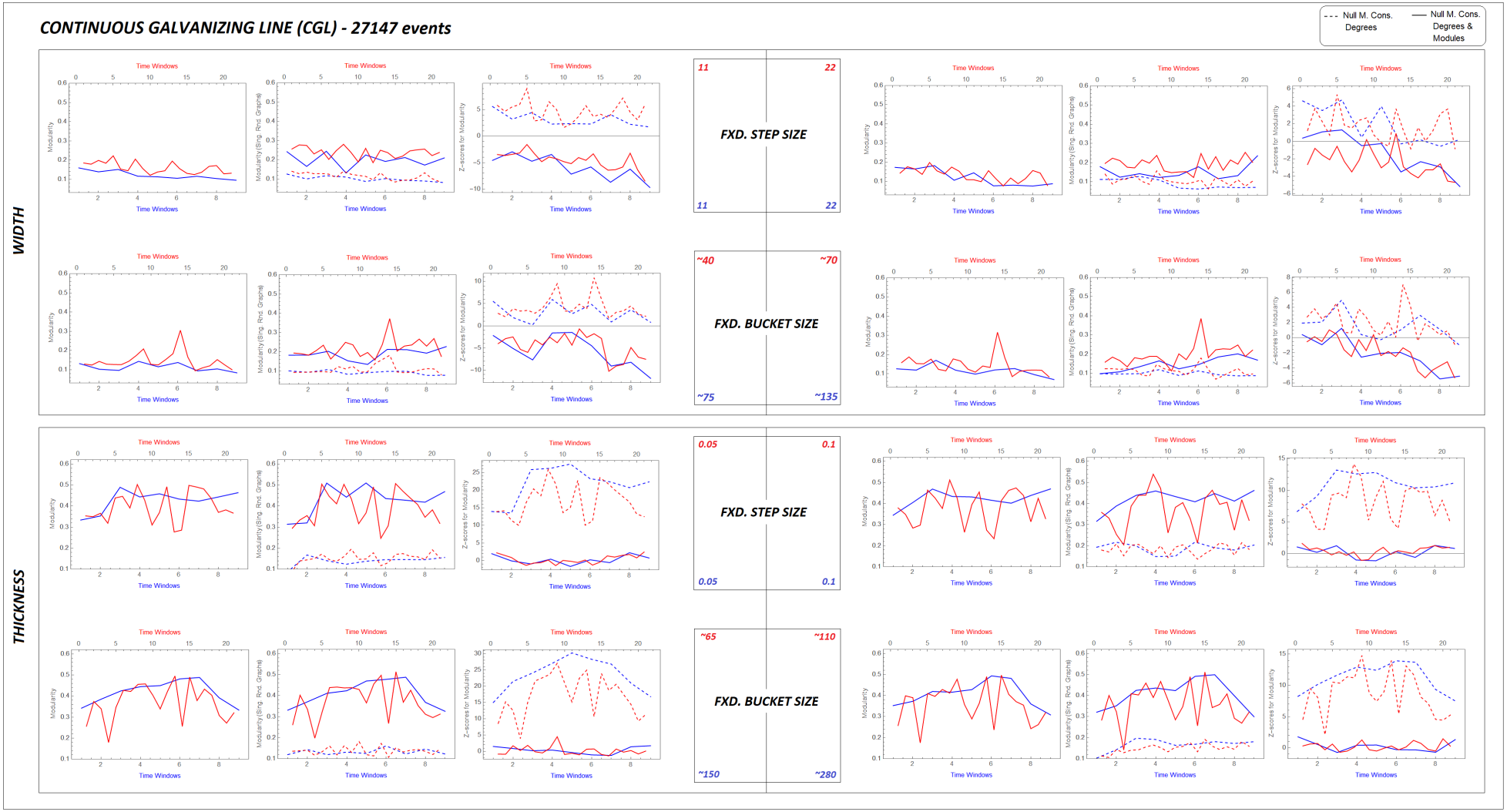


Figure S9: CGL Production Line Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

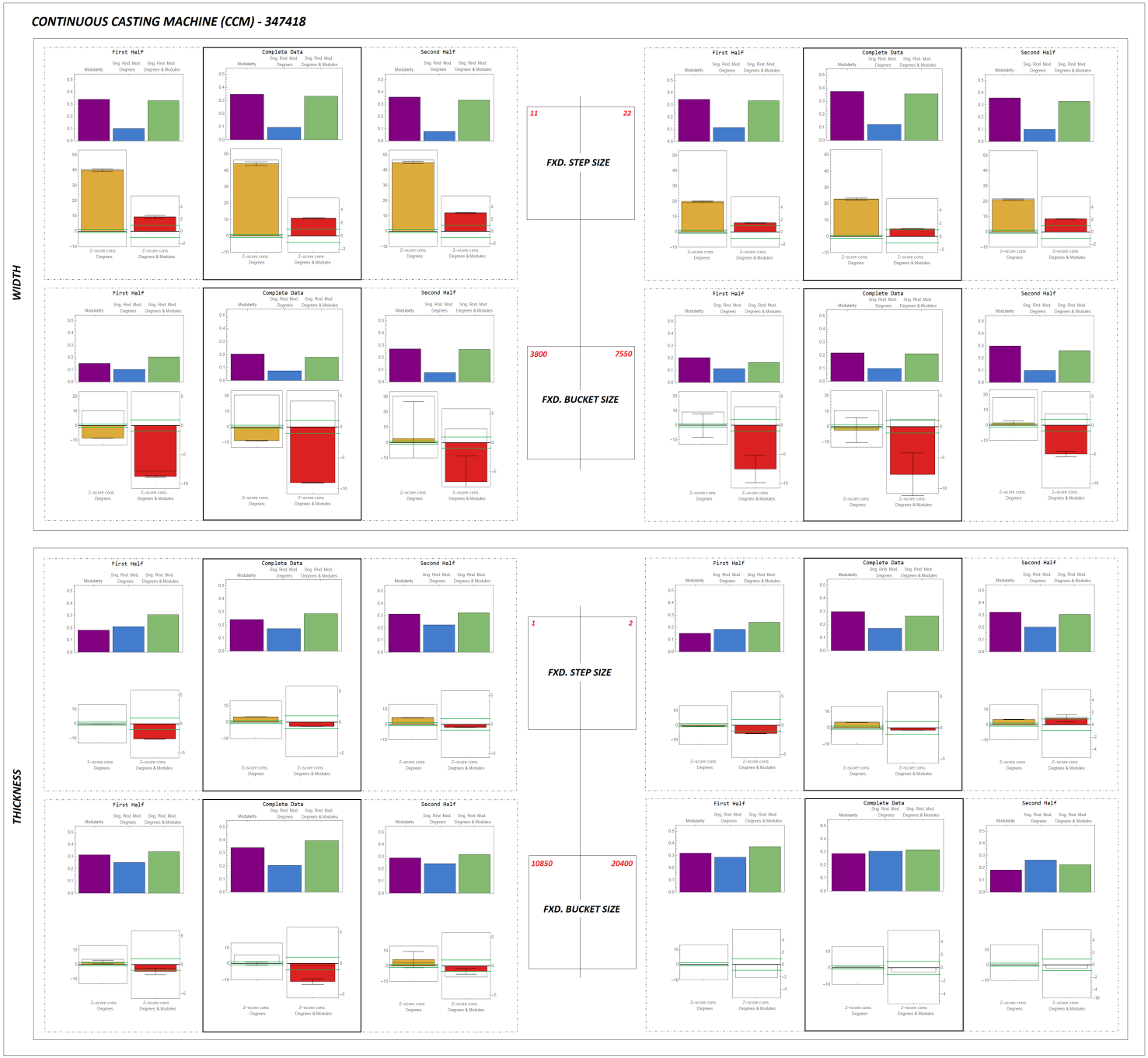


Figure S10: CCM Production Line Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

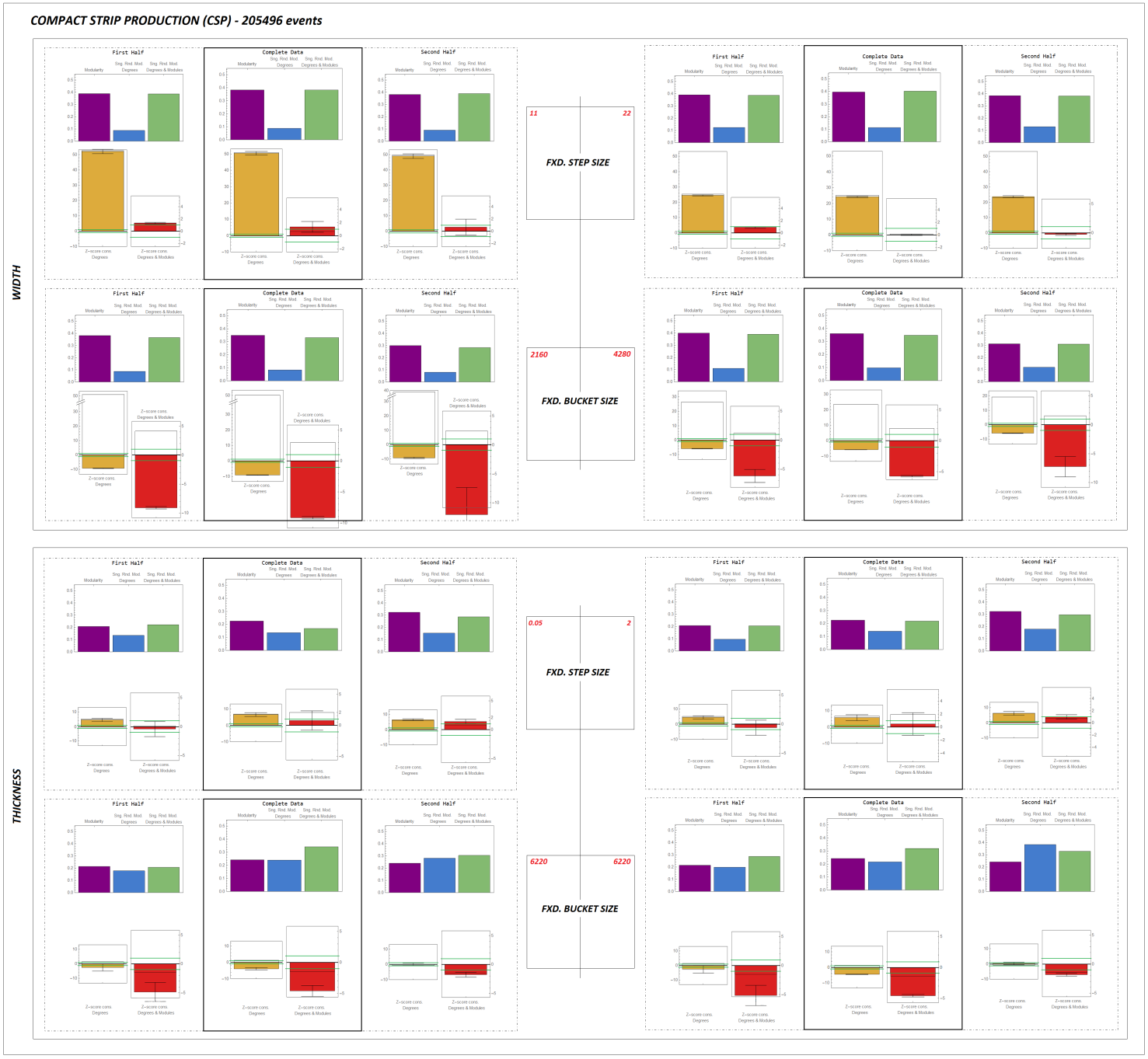


Figure S11: CSP Production Line Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

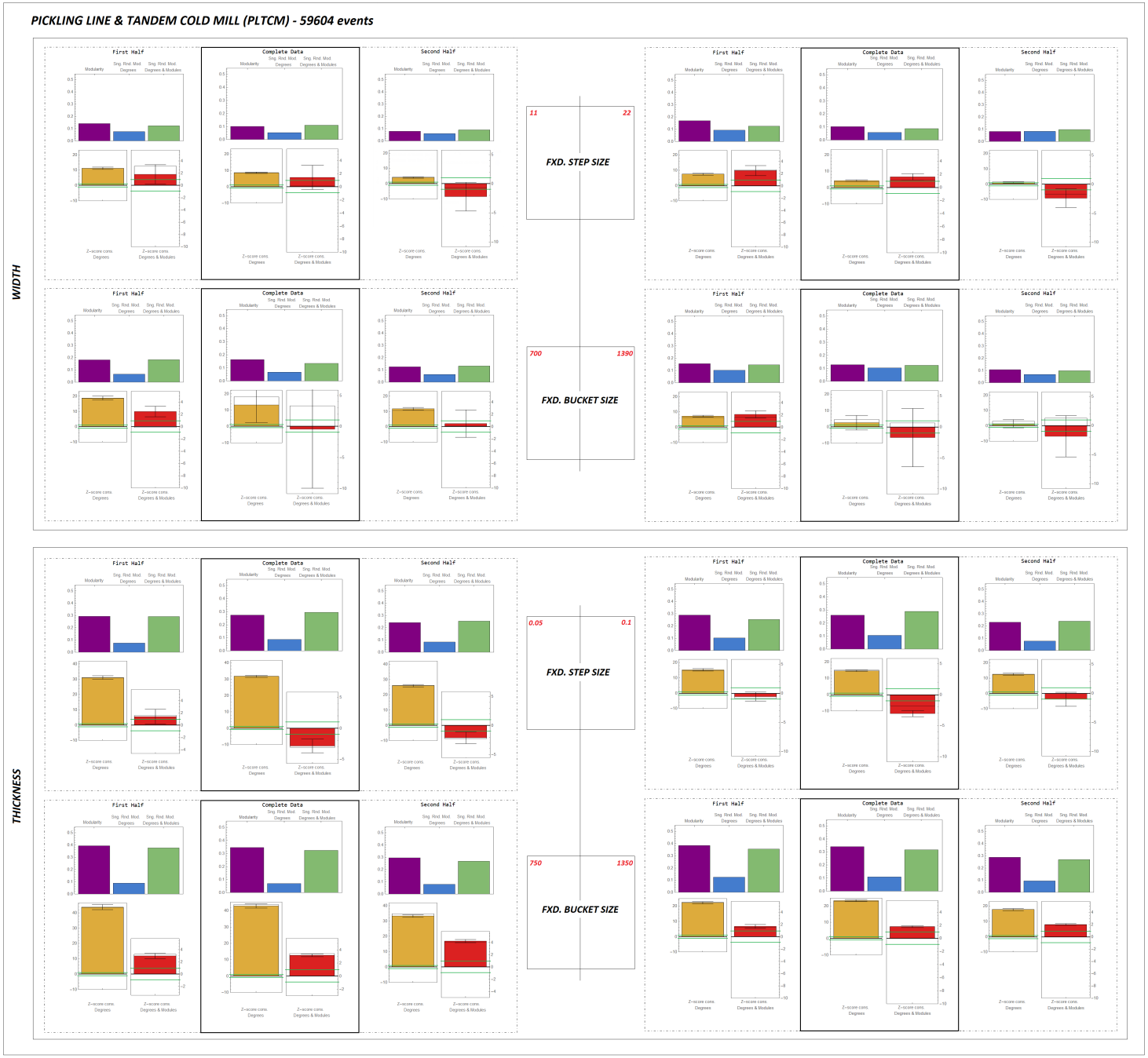


Figure S12: PLTCM Production Line Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

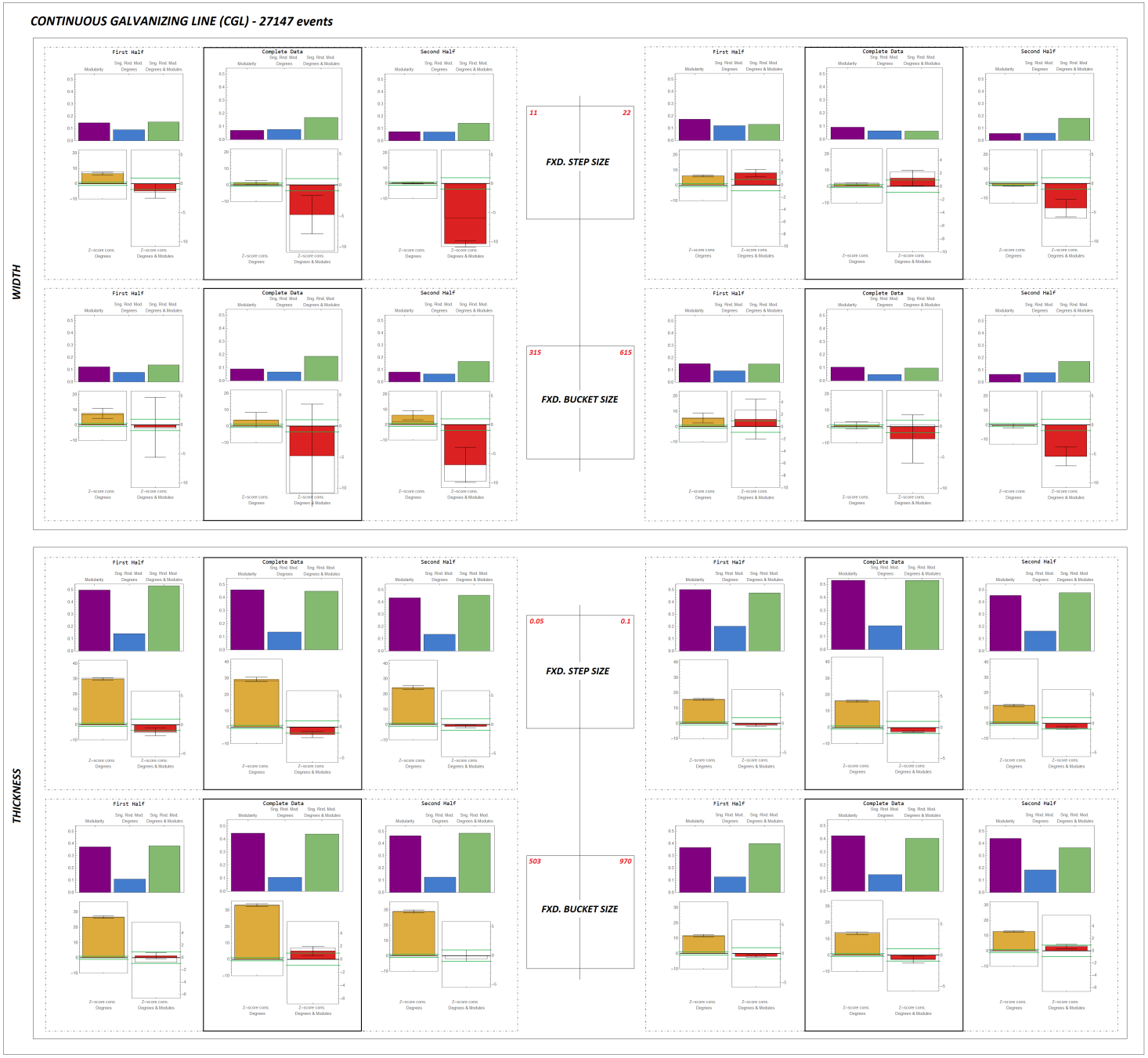


Figure S13: CGL Production Line Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

```
In[17]:= Table[Length@BiomassSeries[[i]], {i, 200}]

Out[17]= {1001, 834, 590, 836, 5, 278, 215, 710, 158, 283, 838, 730, 523,
  290, 768, 690, 320, 280, 123, 691, 752, 709, 906, 901, 1000,
  112, 116, 226, 959, 399, 416, 295, 922, 793, 342, 71, 395,
  246, 380, 873, 915, 994, 462, 908, 382, 755, 153, 608, 15,
  975, 256, 256, 860, 844, 979, 486, 557, 903, 879, 306, 488,
  32, 580, 283, 206, 61, 936, 481, 379, 685, 218, 284, 833, 856,
  247, 825, 490, 870, 37, 169, 245, 257, 654, 175, 138, 691, 72,
  598, 274, 645, 741, 669, 911, 24, 400, 679, 392, 38, 87, 437,
  809, 253, 750, 704, 12, 902, 646, 554, 478, 907, 269, 555, 318,
  970, 906, 682, 340, 370, 625, 931, 907, 524, 357, 653, 392,
  512, 346, 788, 258, 773, 538, 560, 912, 442, 352, 295, 253,
  436, 967, 490, 591, 778, 474, 95, 409, 626, 131, 728, 163, 375,
  867, 286, 459, 870, 644, 420, 833, 990, 746, 389, 750, 963, 17,
  933, 159, 234, 276, 869, 859, 843, 815, 573, 328, 987, 388,
  857, 385, 910, 350, 607, 612, 284, 372, 375, 249, 596, 240,
  623, 427, 532, 702, 885, 84, 946, 154, 590, 417, 224, 844, 110}
```

Figure S14: Counts of Biomass Series Created for 200 Sequences.

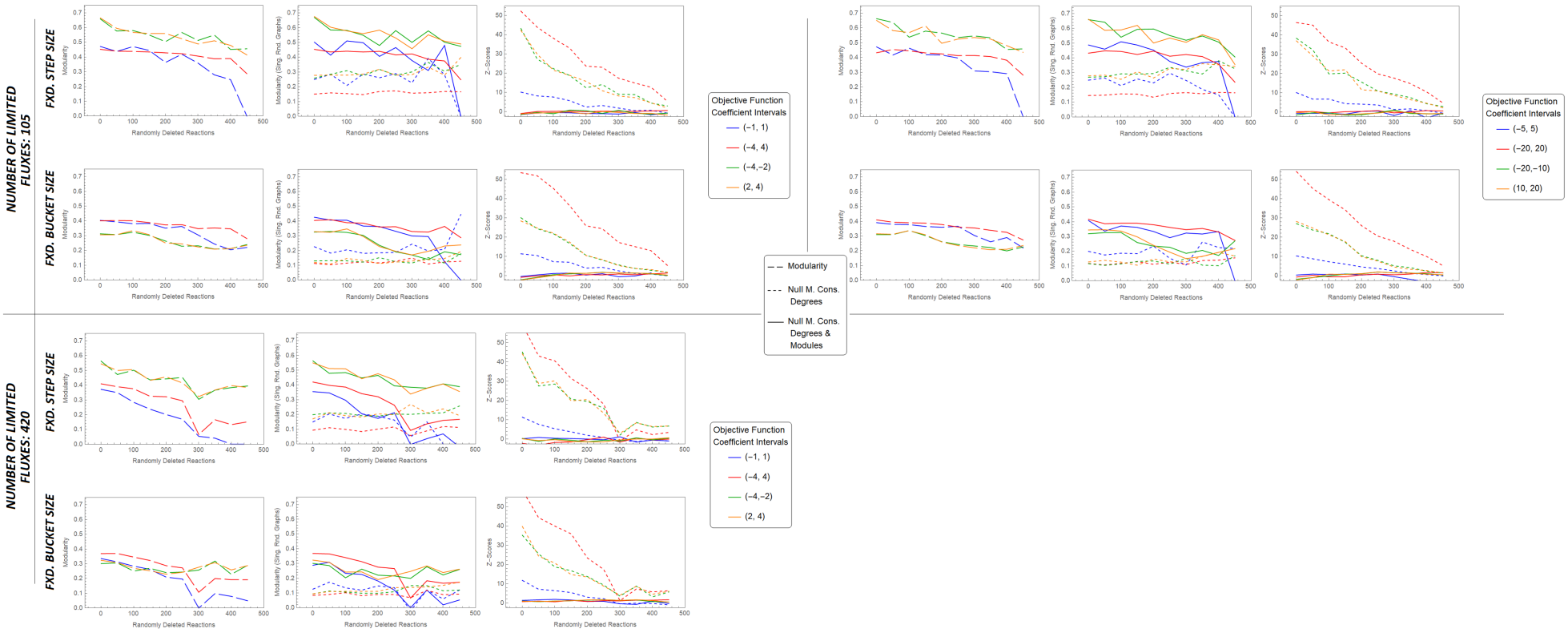


Figure S15: Simulation Results with Initial Terms in Objective Functions.

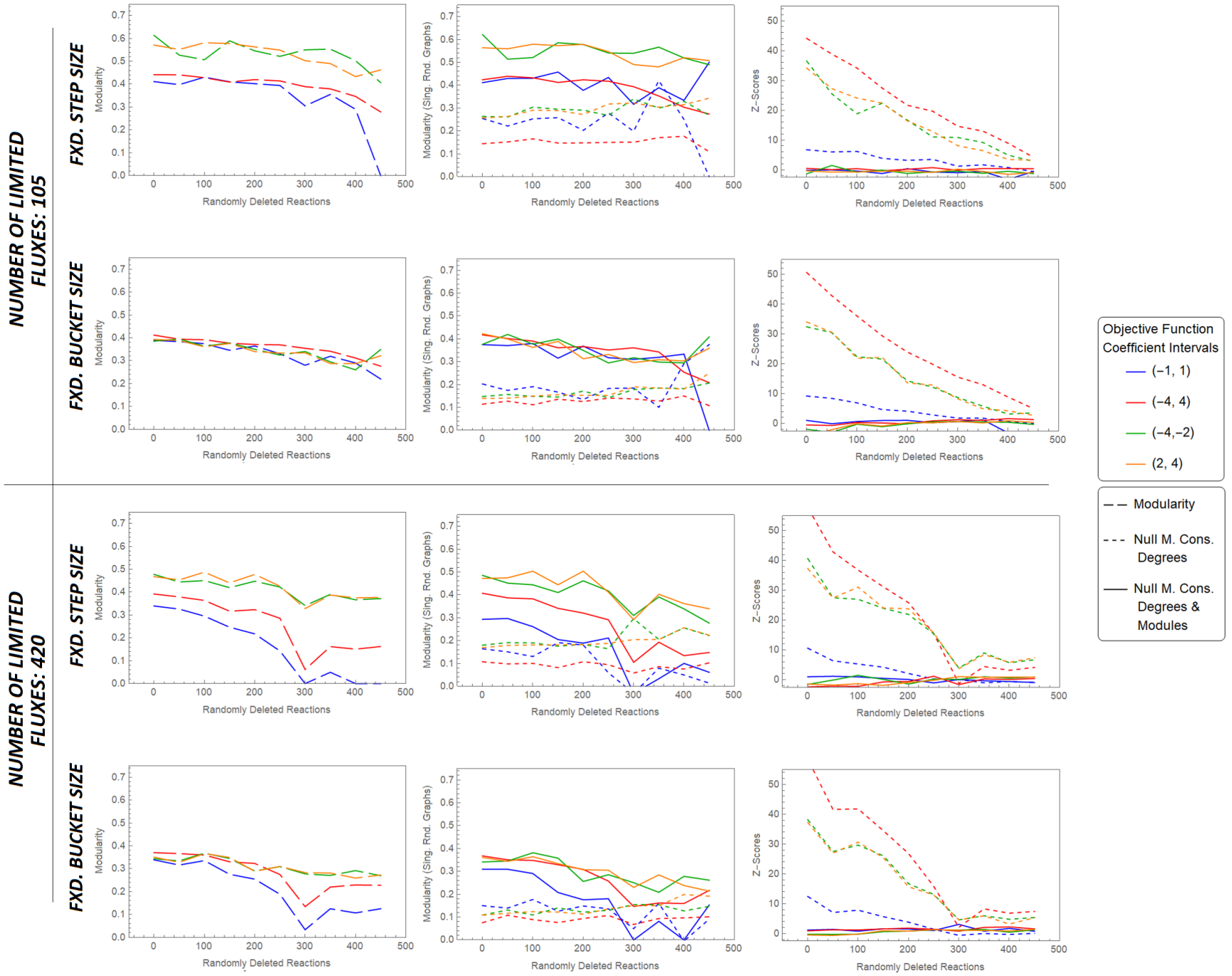


Figure S16: Simulation Results with 25% Reduced Terms in Objective Functions.

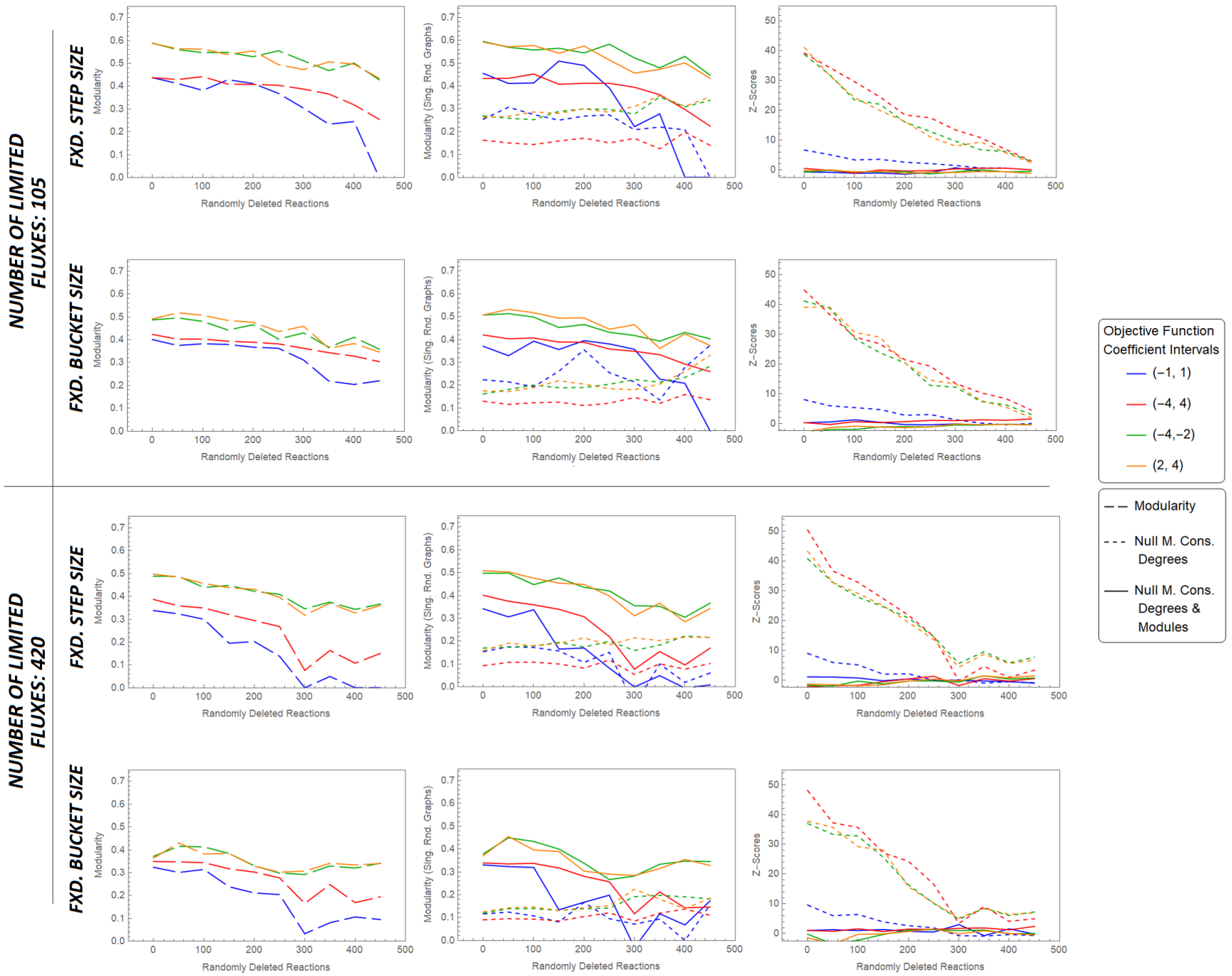


Figure S17: Simulation Results with 50% Reduced Terms in Objective Functions.

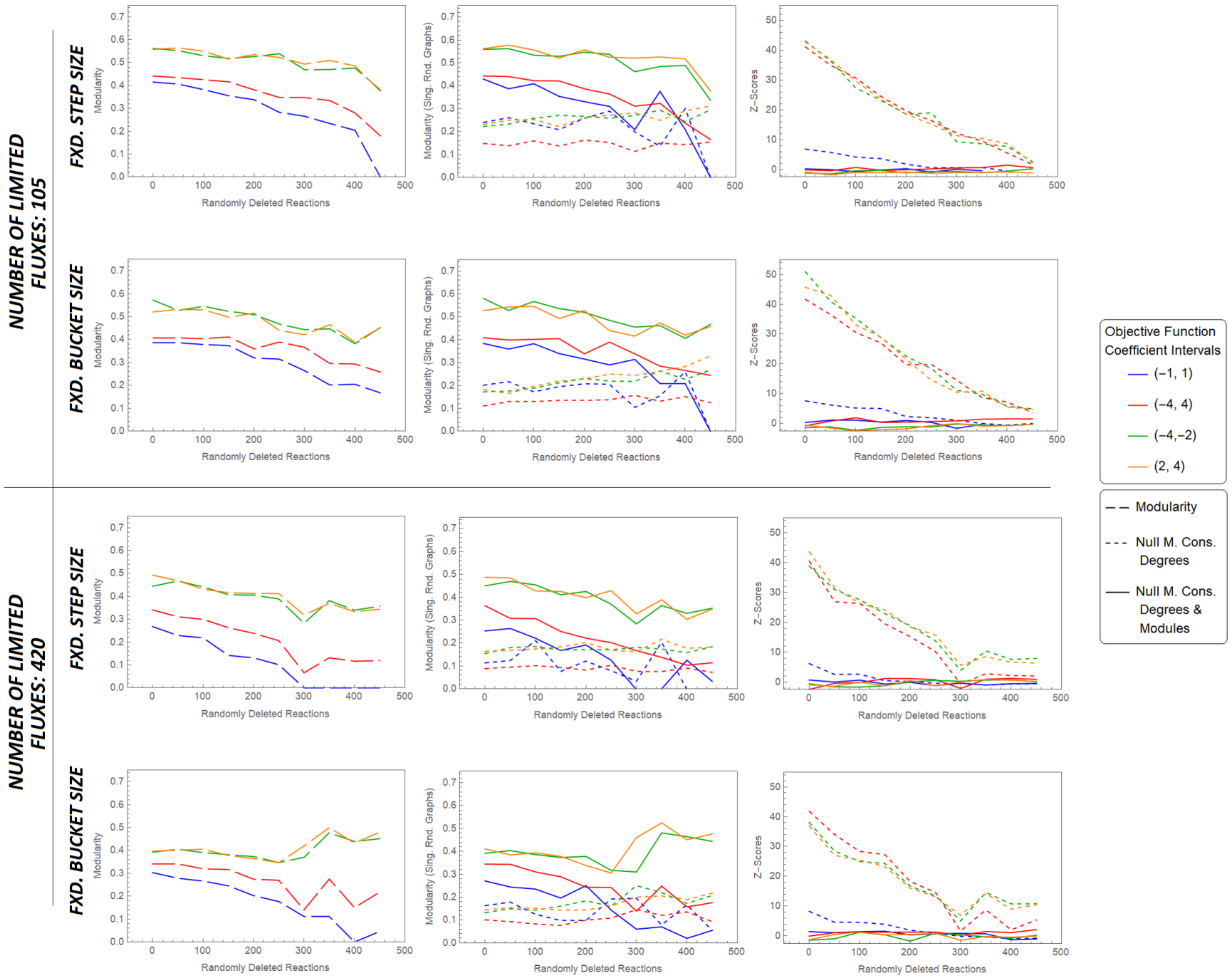


Figure S18: Simulation Results with 75% Reduced Terms in Objective Functions.