



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:

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.

.....
Date, Signature

Abstract

Give a concise synopsis of the work, emphasizing the conclusions; you need not include the supporting arguments for the conclusions. It should be an accurate overall view of the work without needing to read it. State the subject of the paper immediately followed by a summary of the experimental or theoretical results and the methods used to obtain them.

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. This will be achieved by analyzing the statistical properties of association networks over time in a large data set from steel manufacturing. Based on these results, an abstract theoretical framework will be developed to better understand the connection between each type of constraint and the statistical patterns created by it.

Acknowledgements

Contents

Abstract	iii
Acknowledgements	iv
1 Introduction	1
1.1 Background and Motivation	1
1.2 Research Objective	3
1.3 Research Plan and Thesis Organization	4
2 Methodology	6
2.1 Real-life Events	6
Association Networks	6
Binning Methods	8
Network Metrics Analysis	10
2.2 Simulation Model	14
Resource Utilization	18
Production Portfolio Diversification	19
2.3 Integration of Concepts	19
3 Implementation, Analysis and Results	22
3.1 Real-life Events	22
Data Collection and Cleaning	22
Analysis Steps and Results	23
3.2 Simulation Events	27
Design & Analysis Steps and Results	27
4 Conclusion and Outlook	30
5 Bibliography	31
Lists of Figures and Tables	34
List of Equations	35
Supplementary Materials	36 – 53

1 Introduction

1.1 Background and Motivation

A modern steel manufacturing facility has a complex organization consisting of various workstations and can handle different jobs simultaneously. Facility systems are based on Computer Integrated Manufacturing (CIM) implemented in workstations in an automated fashion with combined computer control and digital information [1]. The sensory information received from workstation equipment is stored as data within a server to be incorporated with planning progress to provide functionality, adaptability and effective resource allocation in manufacturing processes [2].

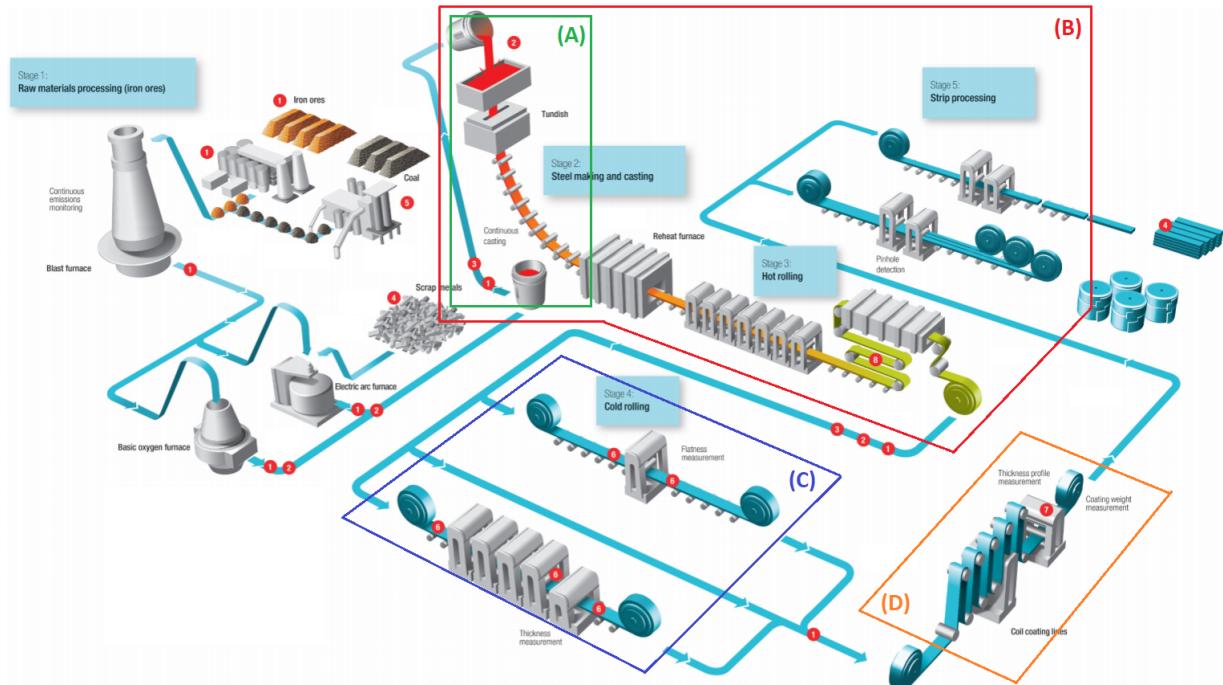


Figure 1.1: Steel Manufacturing Steps [3].

Fundamental steel manufacturing steps with respective production units are shown in Fig. 1.1. Raw materials are melted in the blast, electric arc or basic oxygen furnaces to obtain liquid iron. Accordingly, liquid steel alloy is sent

to the continuous casting line, where it is poured into a mould cavity until it cools and solidifies. The solid material is sent to a further step, the rolling process, which can be performed in two different modes: hot rolling and cold rolling. It allows obtaining the material's desired mechanical properties, uniform thickness, a control on width dimension, and converting material to a flat and rectangular slab, a semi-finished steel product. The solid material is heated in a reheat furnace before it's pressed in the hot rolling unit. The cold rolling unit improves surface finish and flatness, and it allows to modify metal work hardening. The output slabs of the rolling process might be converted into compact coils featuring high lengths unless they won't be sent to further process units on the continuous production line. Pickling is a treatment to remove rust and impurities on the slab surface; it is applied before cold rolling processes and makes it easier to work on the material. Hot-dip galvanizing process is an effective coil coating technique. Galvanizing is an application of protective zinc coating on the steel surface to improve corrosion resistance.

The above-explained processes and workstations can be arranged in various compact plant solutions based on the requirements of different facility organizations or demands. Table 1.1 shows four separate production lines that the SMS group supplied to the steel manufacturing facility: Big River Steel, located in Arkansas, USA [4].

Label in Fig. 1.1	Production Unit / Plant	Description
A	Continuous Casting Machine (CCM)	Mould steel cools and solidifies, passing through the mould cavity.
B	Compact Strip Production (CSP)	Compact plant including CCM, reheating furnace, hot rolling unit and strip processing unit.
C	Pickling Line & Tandem Cold Mill (PLTCM)	Compact plant including a turbulence pickling section and a tandem mill.
D	Continuous Galvanizing Line (CGL)	Application of protective zinc coating on the steel surface to improve corrosion resistance.

Table 1.1: Production Lines Including Various Machine Modules and Compact Solutions

Each process has its own technical or physical constraints considering the above-explained production steps. Constraints arise from those technology-driven production types related to facility capabilities [5].

Optimizing individual sequences in each process is a necessity for a successful local unit. Local constraints on different production units are integral parts of a global optimization problem that human expert planners tackle for a solution. The sequences produced in production lines are available as data output, so-called 'imprints' of what has been on sequence designers' minds. Therefore, looking at the historical production data and investigating the properties of those order sequences that have already been produced give indirect access to the patterns related to human experts' knowledge systems. The technical and physical constraints mentioned in the introduction section are unique to those specific production lines, and they vary under differently customized production lines. Produced order properties such as thickness, width, temperature, and chemical composition are the characteristic features of order products that are possibly shaped under the effect of those constraints. An investigation on the features of orders in the identical production sequences and comparing them might give interesting clues about related constraints and correspondingly provide patterns about human experts' decision strategies.

I should discuss and refer to different constraints from the literature introduced for the manufacturing life cycle.

As a motivation, I need to introduce different categories of constraints as technical constraints, performance-indicator based constraints to be quantified in the context of the FBA in the further steps of this work.

Are logistic constraints, physical and chemical constraints coupled to topological features of the association network?

1.2 Research Objective

Our hypothesis: different types of constraints create non-random features in the association networks for different binning schemes. Networks derived from various types of binning. Do they show non-random features when I have performance constraints or other types of constraints?

Two fundamentally different constraints acting on the manufacturing process: technology-driven constraints and load-driven constraints, were shaped hypothetically and defined as two distinguished network approaches: fixed step-sized and fixed bucket-sized networks.

FSS graphs had high modularity. That means that the actual quantity I discretize creates the constraints, while in the case of FBS, it would be the volume of orders that makes my constraints. That summarizes our hypothesis.

Explanation of my hypothesis is a theoretical/conceptual framework as a start-

ing point for the investigation. It is a well-defined valid object and based on facts. Moreover, it is structured to discriminate the two types of constraints in the statistical properties of the production data.

The initial step of this master thesis work was to quantify the characteristics of two hypothetical types of constraints in industrial production: technology-driven constraints and load-driven constraints.

1.3 Research Plan and Thesis Organization

Having a small simulation framework would help understanding mechanistically the difference between the two types of constraints. But data analysis would provide us some intuition on what to look at. How can even detect that some real data are impacted/shaped by these two types of constraints?

If I could now, in some way, do this analysis in time windows across 2-3 years of production data, then I undoubtedly will encounter time windows where the load in the system was really high so I would expect that the other type of constraints also place a role.

It is plausible either the tech constraints do not go away, the other type of constraints is present with varying strength/importance in the system.

Optimization principles coming from Operations Research:

Considering the system as a linear set of fluxes of material flows. Finding the pattern of material flow that maximizes the output. The advantage of this approach: the notion of constraints is already present in that framework.

Can these constraints be properly discriminated? Is there a formal way of defining them and are there functional consequences? Do they impact how the system behaves? Can I distinguish the signature of these constraints in data?

Modeling alternatives: OR Methods, Discrete Event Simulations

Methods are introduced here as indicative of two fundamentally different constraints acting on the manufacturing process: technological constraints on the one hand and constraints related to material flow and production capacity on the other.

I plan to quantify the characteristics of two hypothetical types of constraints with an Operations Research Model consisting of two steps. First, analyzing the statistical properties of association networks over Time in an extensive data set from steel manufacturing; second, developing an abstract theoretical framework to understand better the connection between each type of constraint and the statistical patterns created by them.

Formulate the binning methods here because this will describe the hypothesis underlining my thesis.

My Operations Research Model (OR model) combines Steel Manufacturing Events Analysis and Flux Balance Analysis. The art form of this model is to structure a standard data format and a shared analysis logic that allows comparing the results from manufacturing data and simulation data.

2 Methodology

Introduce proposed concepts in the OR model.

Here is a brief introduction to Association Networks, Modularity as a Complex Networks metric, Randomization with Null Models and Flux Balance Analysis (FBA).

Introducing the analysis of the real-life events with the related concepts and explain the relation of the thesis hypothesis with this analysis pipeline.

Here is an explanation for generating a data structure with OR-modeling in the combination of those. Usage of linear programming and creating sets of synthetic data allow comparing the statistical characteristics of their association network with those formed from the real-world data set from steel manufacturing.

Mention that more detailed information will be given in the following sections, guide the readers who know Association Network and FBA concepts to the Applications and Results Chapter.

2.1 Real-life Events

Association Networks

Beyond a simple network graph representation of 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 [6]. The relevant 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 manufacturing 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:

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

Table 2.1: Arbitrary Manufacturing Data Set D .

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 knowledge is not a simple task for large and complicated real-life manufacturing 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 [6]. 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

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

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.

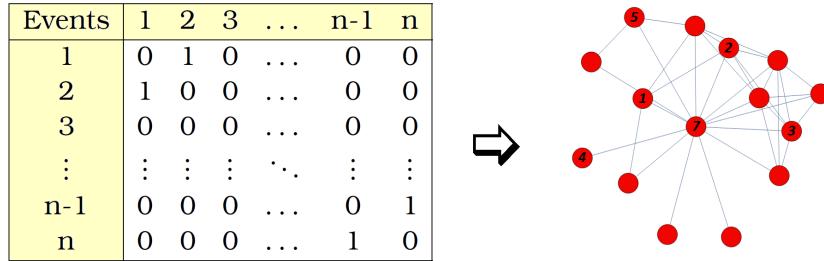


Figure 2.1: An Arbitrary Representation for Adjacency Matrix and Its Graph.

Binning Methods

The data set D events can be labelled with a typical value interval (the so-called binning size) for every Feature-A value with a slight difference to each other. Binning generation for the events allows us to investigate them in a **sequential manufacturing system** and can be performed in alternative ways.

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.

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

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

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

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

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

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 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.

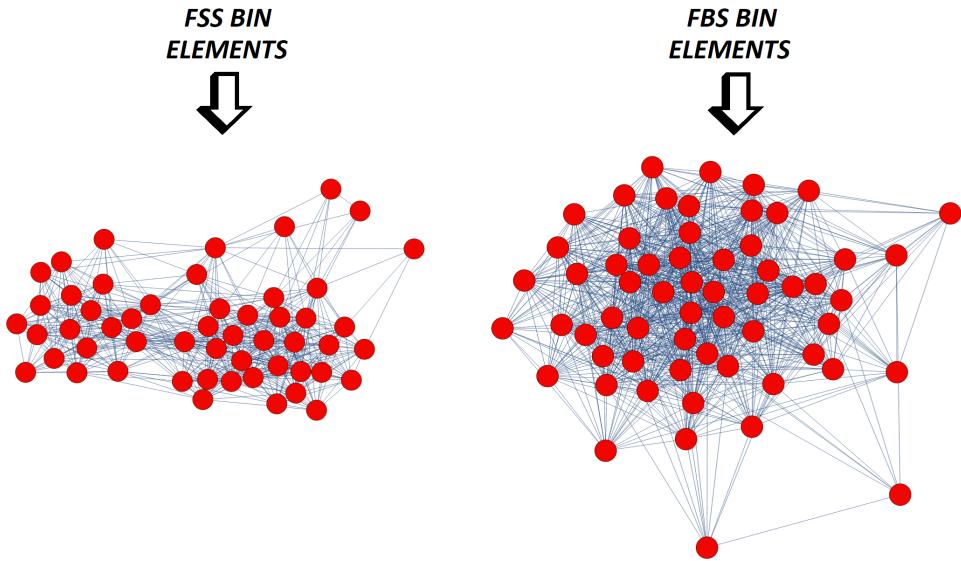


Figure 2.2: Graph Results For Two Different Network Approaches.

FSS and FBS networks generation for the production events underlie the developed hypothesis of this thesis work: Non-random features of the association

networks derived from these two methods.

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 [7]. 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 [8]. Communities (the so-called modules) are groups of nodes that probably play similar roles within the graph [9]. 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

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

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 maximize Q . After first partitioning, the edges falling between

the further divided sub-graphs are neglected, leading to a wrong maximization quantity. For this reason, the author introduced the additional contribution ΔQ . [10]

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 [11]. The distribution of degrees in random graphs is highly homogeneous, and they do not reveal a significant level of order or organization [9]. 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 [6, 9, 12].

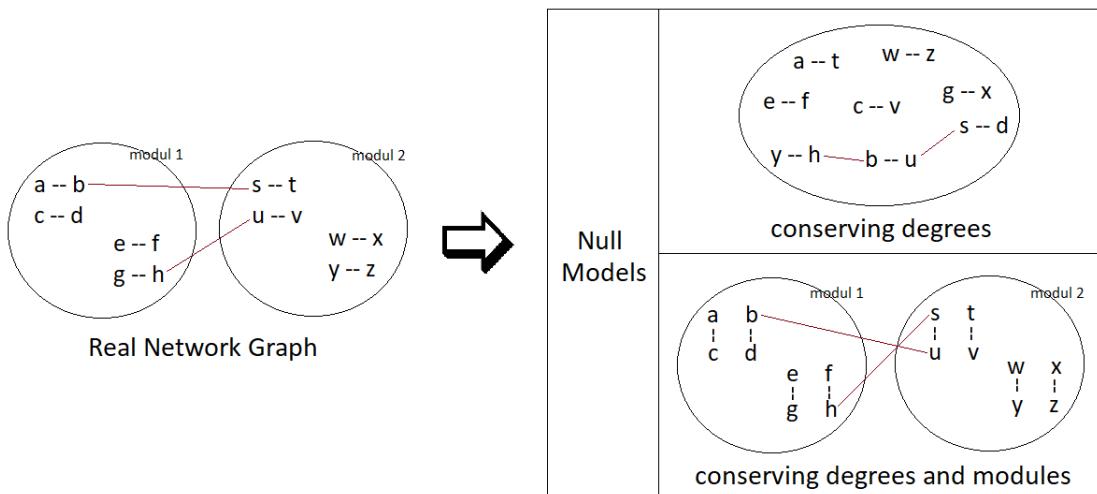


Figure 2.3: Formation of Different Null Models.

Our analysis pipeline considers two types of randomized 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 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 [9, 13]. In NM-m, intra-edges

and inter-edges among modules are shuffled separately by preserving the original degrees sequence [13]. We should emphasize 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.

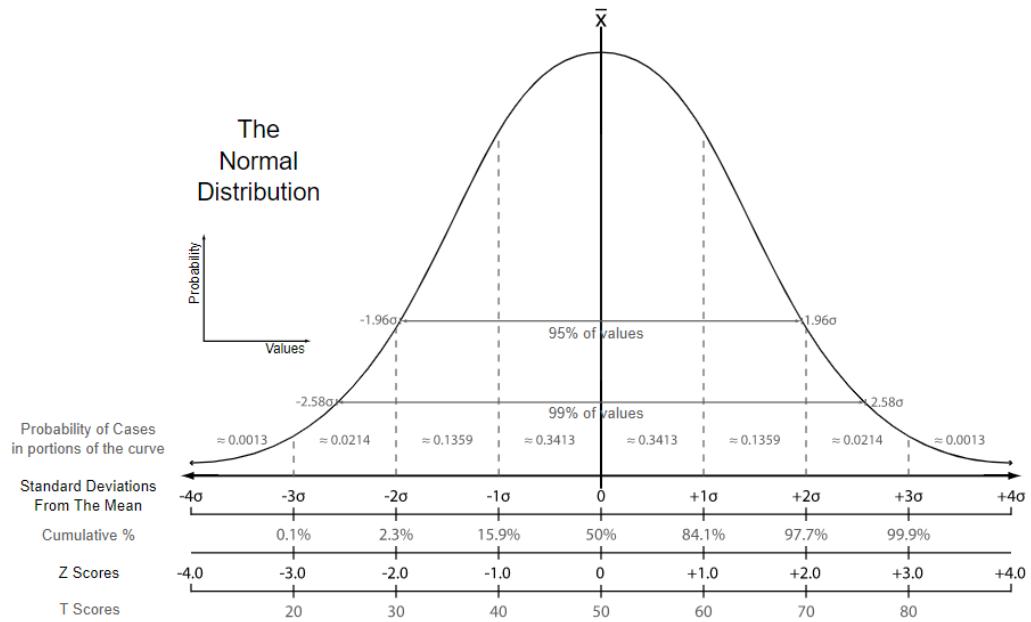


Figure 2.4: Chart Comparing the Various Grading Methods in A Normal Distribution [14].

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. 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

$$z = \frac{x - \mu}{\sigma}. \quad (2.3)$$

x is the modularity value for the real network; μ is the expectation value (mean) of x in the set of 1000 randomized graphs. σ is the standard deviation of x in the randomized 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 values lower than 1 or higher than -1 indicate that the real network is incidental; the z -score values between 1 and 2 or between -2 and -1 suggest that the real network is close to random characteristics. In contrast, the z -score values 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 randomizing 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 organized by following a hierarchical rule [15] form large groups displaying a high degree of clustering while the degree distribution follows a power law [16]. That hierarchical organization of nodes creates a nested modularity structure in the networks, having modules within modules. That type of organization is observed in several real networks like the World wide web, the Internet at the domain level, actor-network [16], macaque & cat cortical systems [17] and the Escherichia coli metabolic network [18]. Suppose the z -score is greater than or equal to 2 using NM-m. In that case, we assume that the real network has a complicated structure or hierarchical organization since the randomizing scheme would destroy the internal modularity of graph modules.

We pretend that the real network structure is more homogeneous than it is and pretend that it is statistically reliable and not distorted or skewed. However, those presumes are slightly wrong, and those can lead to cases sometimes looking statistically significant even though they are not. To distinguish the reliably distributed network from the skewed or distorted ones, we assume the z -scores less than or equal to -2 concerning NM-m as an indicator of asymmetry in the real network, in other words, having significant differences in the number of inter-module links. In this case, some modules are tightly connected while the other modules are sparsely connected.

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

Real Network Z-score	Null Model	Network Structure
$z \leq -2$	degrees sequence is conserved (NM-d)	nonrandom, complicated
$-2 < z < 2$		random, not modular
$z \geq 2$		nonrandom, modular
$z \leq -2$	degrees sequence and modules are conserved (NM-m)	complicated, distorted/skewed-modular
$-2 < z < 2$		not complicated structure
$z \geq 2$		complicated, hierarchical organization

Table 2.4: Expected Network Structures With Respect to Null Models.

2.2 Simulation Model

The genome-scale integrated networks are necessary tools used by metabolic engineers on model design, theoretical and computational analysis to understand how the biological system of microbial organisms works [22]. 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

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

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. [19]

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, $A_{ij}^m = f(s'_{ij})$ and $A_{ij}^r = f(s''_{ij})$, 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 [20]. 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.

Studying biological metabolic systems and designed models to achieve cellular objectives like cell growth or ATP (Adenosine Triphosphate Production) necessitates various tools to be integrated with reconstructed genome-scale networks [21, 22]. One of the commonly used tools is Flux Balance Analysis (FBA) as an optimization 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 [23, 24].

While one can express the metabolic fluxes in a one-dimensional array (the so-called flux vector V) as

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

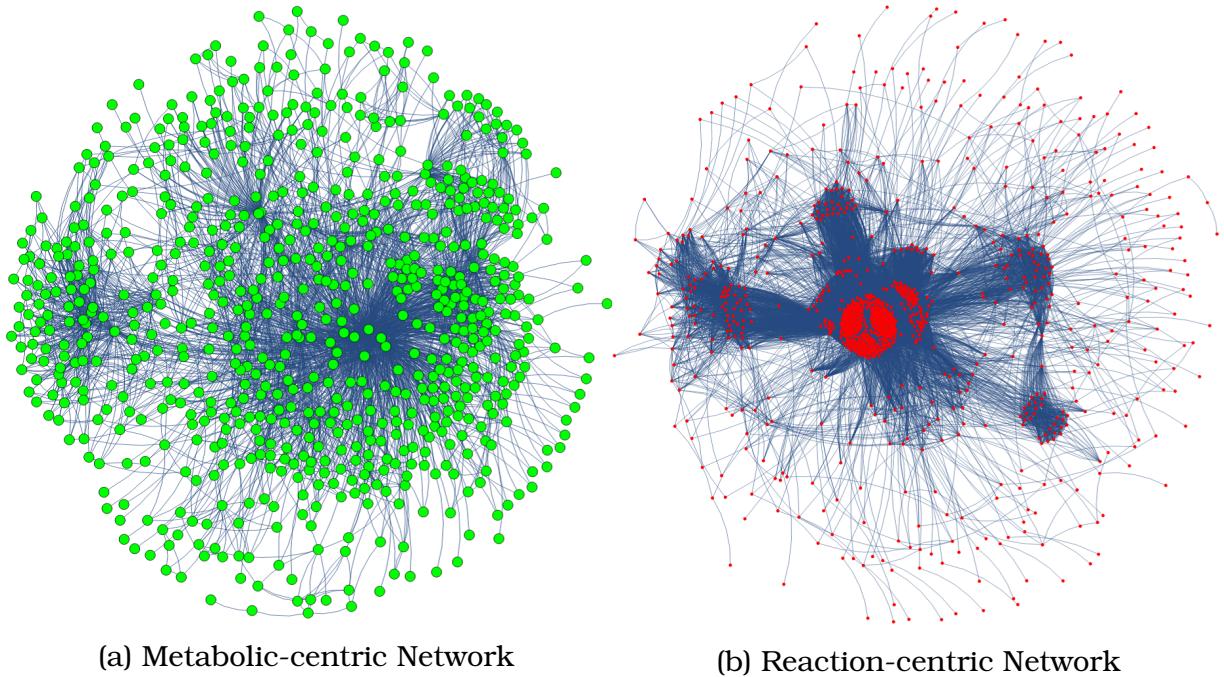


Figure 2.5: Network Representations for Homo Sapiens Metabolic Model

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 [23, 24].

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

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 organization 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 [23]. Therefore, one can formulate an optimization approach to identify reaction network steady-states that maximize the biomass [23, 24] or control the production of specific metabolites [25]

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 [24]:

- i. to discover allowable characteristic properties in the genome-scale network reconstruction,
- ii. to mimic probable physiological functions like biomass or ATP production 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

$$O = [o_1 \ o_2 \ \dots \ o_r] = (o_i) \in \mathbb{R}. \quad (2.7)$$

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.

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

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 optimization 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 optimization process.

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 [24]. The optimization result: flux vector V (Eq. (2.5)) maximizes the objective function in the form of a flux distribution [23, 24]. 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.

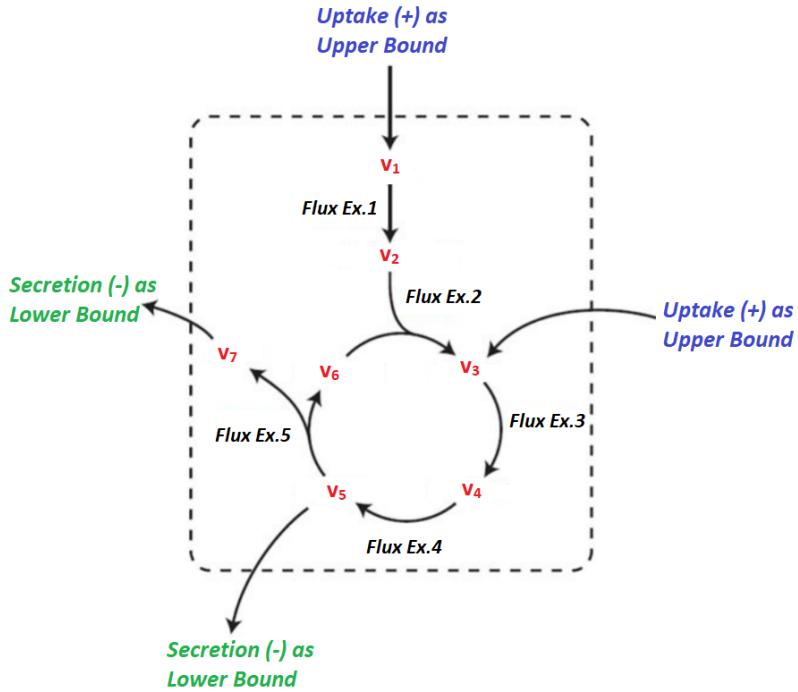


Figure 2.6: A Simplified Reaction-centric Network Sketch Shows The Reactions for Exchange, Uptake and Secretion.

Different solution vectors of V can be obtained from the linear optimization 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 [26]. 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 optimization is carried on.

Resource Utilization

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. [24, 27, 28, 29]

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

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 optimization.

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

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 [26]. 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.

Production Portfolio Diversification

The objective function can be assumed as a production plan that rules the diversity of products that metabolism takes into account to maximize cellular growth [26]. 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 approach.

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 **real-life events**. 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.

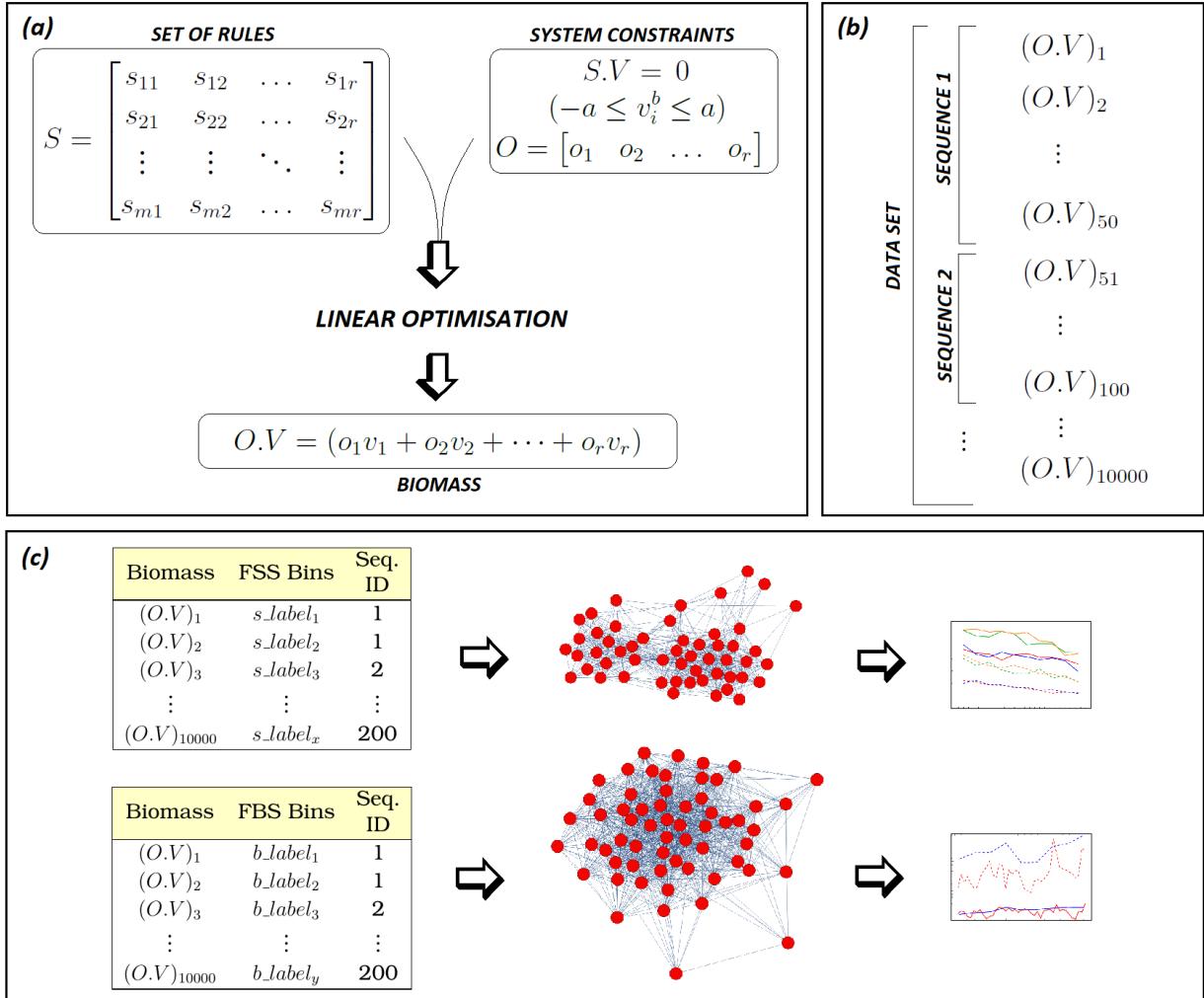


Figure 2.7: Simulation Model Illustration

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 **real-life production event**. The optimisation algorithm is run 10000 times to create a data set with 10000 events.

Fig. 2.7b shows the data structure generation by introducing a production

sequence concept. **Each sequence shows consistency among its production events**; 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 fluxes 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

Investigating constraints impact in time windows was performed by analysing two different association networks; FSS-n and FBS-n.

Those two different types of networks were applied in all ten time-window, and average modularity metric plots were generated.

3.1 Real-life Events

Data Collection and Cleaning

The queries in Structured Query Language (SQL) were generated to find and pull the production orders data from the database across 2–3 years of production work completed in the production lines, introduced in Table 1.1. The SQL queries are given in the Supplementary Materials: S1, S2, and S3.

At the beginning of the data cleaning process, the raw data was handled considering the string-type data values conversion 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 consider usable parts and fill the gaps in the data sets.

- The steel material density was considered between 6.5×10^{-6} kg/mm³ and 8.5×10^{-6} kg/mm³. The production orders with density values out of that range were discarded from consideration.
- The production feature, length values were taken into account with millimetre (mm) units in the metric system.
- Machines input capacity limit ranges were identified for the production features; width, thickness, and weight as 800–2000 mm, 40–90 mm, and 2669–26690 kg.

Considering the preconditions mentioned above and $\text{density} = \text{mass/volume}$ equality, 0 values were replaced with the calculated values in every production order with a maximum of one unknown value from the features; width, thickness, weight, and length. **Production orders (the so-called events)** with two missing values were compared with consecutive events, and missing terms were filled based on the consistency among the same sequence events. Sequences with less than 50 events were removed from the data sets considering those short sequences might be generated for some test processes. 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: 64, 026 events.
- CGL data set: 31, 230 events.

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 the continuous production line, as mentioned in **the first section of the Introduction**.

Analysis Steps and Results

The complete analysis of real-life 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 in the given order considering **the production portfolio evolves from producing products in a diverse range to producing specialised products from CCM to CGL**. 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.

Two fundamentally different constraints: technology-driven constraints and load-driven constraints are acting on the manufacturing process. Alternative binning methods and different network approaches were identified 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. Resulted z-scores are more straightforward than the resulted 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 reveals 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 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 amount. 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.

Condensed analysis results as bar charts are given in Fig. 3.1, 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. For each of the z-scores, error bars were included in colour as 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. Green lines are indicated on the values: +1 and -1 as the significance thresholds for signals as one standard deviation range.

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 have complicated and distorted/skewed-modular structures 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.

For PLTCM and CGL, FSS and FBS networks for the thickness feature are modular and do not comprise any complex textures; therefore, that case is more reliable. FSS and FBS networks are getting complicated and distorted for the width feature, considering NM-m z-scores less than -2 and error bars with long T-shaped symbols.

As a summary of the bar charts in Fig. 3.1, the modular structure of the networks shifts from width to thickness when moving the observation angle from CCM and CSP to PLTCM and CGL. **Two different binning schemes: technological constraints and load constraints, show different results through production lines going from a more general production portfolio to a more specific product portfolio.** Moreover, thickness is a more specialised feature than the width feature since the FSS and FBS networks have more significant z-scores for the thickness feature as the specialisation increases. **Load constraint networks** do not show a significant change in width feature as going further in the production specialisation; however, **technological constraint networks** get closer to randomness and start to get distorted in that direction.

3 Implementation, Analysis and Results

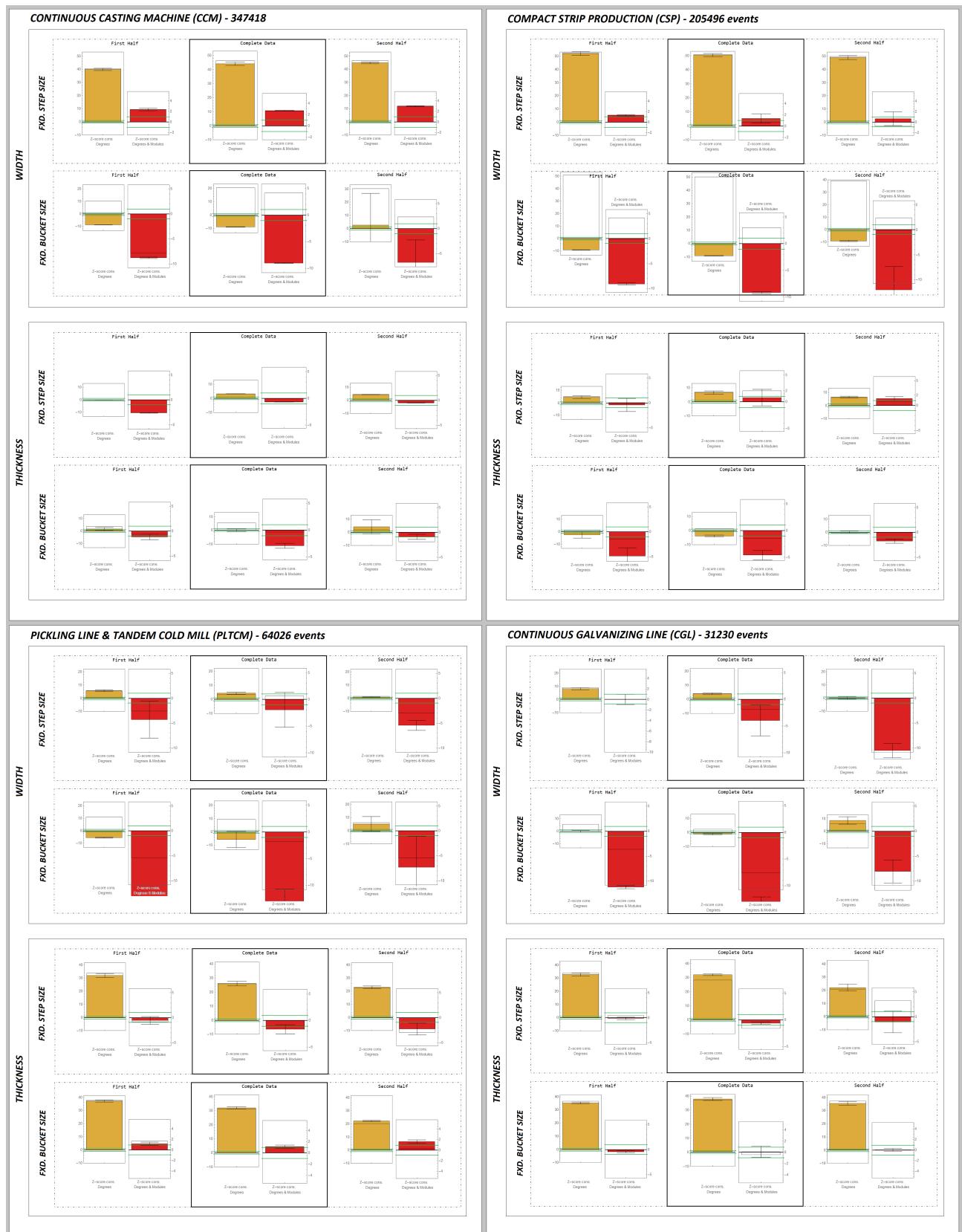


Figure 3.1: CCM, CSP, PLTCM, CGL Analysis Bar Chart Results: Z-scores.

3.2 Simulation Events

Design & Analysis Steps and Results

Briefly explain *in silico analyses* attempts /numerical experiments from the generated data.

Plots in Part-1 of the file belong to association networks of four different synthetically created sequence data sets, and each represented in various colours: green, blue, orange, and red. Part-1 data sets were derived with fixed reaction bounds but with varying coefficients of objective functions. Part-2 also presents plots for four different synthetically created sequence data sets with fixed objective function coefficients but with variable reaction bounds.

Each data set has a length of 10,000 events shared equally in 200 sequences. Randomly picked subsets of fluxes were kept the same within the sequences but having varying coefficients of objective functions.

Limitations on resources were performed in two different ways; first, restriction on upper & lower bounds and second, deletion of fluxes.

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. Randomly chosen 105 fluxes out of 1008 were matched with $(-5, 5)$ as the first step. Furthermore, 105 was doubled (212) and then quadrupled (425). An important detail is that all three sets of choices were done randomly, and they are not added on top of already selected 105. The same three sets of fluxes were used in the computations as restricted bounds in every further step.

Deletion goes in the line: 0, 50, 100, 150, 200, 250, 300, 350, 400, 450. As explained previously, deletion was done by assigning $(0, 0)$ bounds to the fluxes. On the last step, almost half of the total fluxes (1008) were erased.

In an ideal scenario, we would find that association networks derived from the generated data, in the one case; produce high modularity for FBS and in the other case produce high modularity for FSS. Because then we have linked these two data processing schemes to different forms/to different categories of constraints.

We see here that when I vary one constraint about the richness curve plots, I go from a factory that produces anything at random to a more specific factory in their production plans. The modularity at FBS increases, modularity at FSS does not increase. From left to right, I increase the constraint that the production plan (or portfolio of the factory) impose on the whole production process. The production portfolio is grouped into speciality products on the right end. An increase in modularity is only valid for the green and orange

curves, which means that this effect of the changing of the portfolio only takes place. In addition, I impose certain constraints on the material flow (FBS). I enforce the production plans. In some sense, with the coefficients of objective function being symmetric around zero, I allow them also to near zero. I allow for the case this product doesn't take place. The red and the blue curves are less interesting, and they are rather serving as an orientation. We wouldn't expect them to be severely affected by changes in the richness of the objective function because the terms are cancelled out. So the interesting curves are the green and orange one that we can argue for it. We don't expect a strong impact of the richness of the objective functions for these cases (red and blue), and that's indeed what I numerically observed. So it makes sense to have these curves, but the interesting curves are the green and the orange ones. Blue curve z-score is very hard to compute, and we should not trust it that much; also, its z-score curve is lower than the red z-score curve.

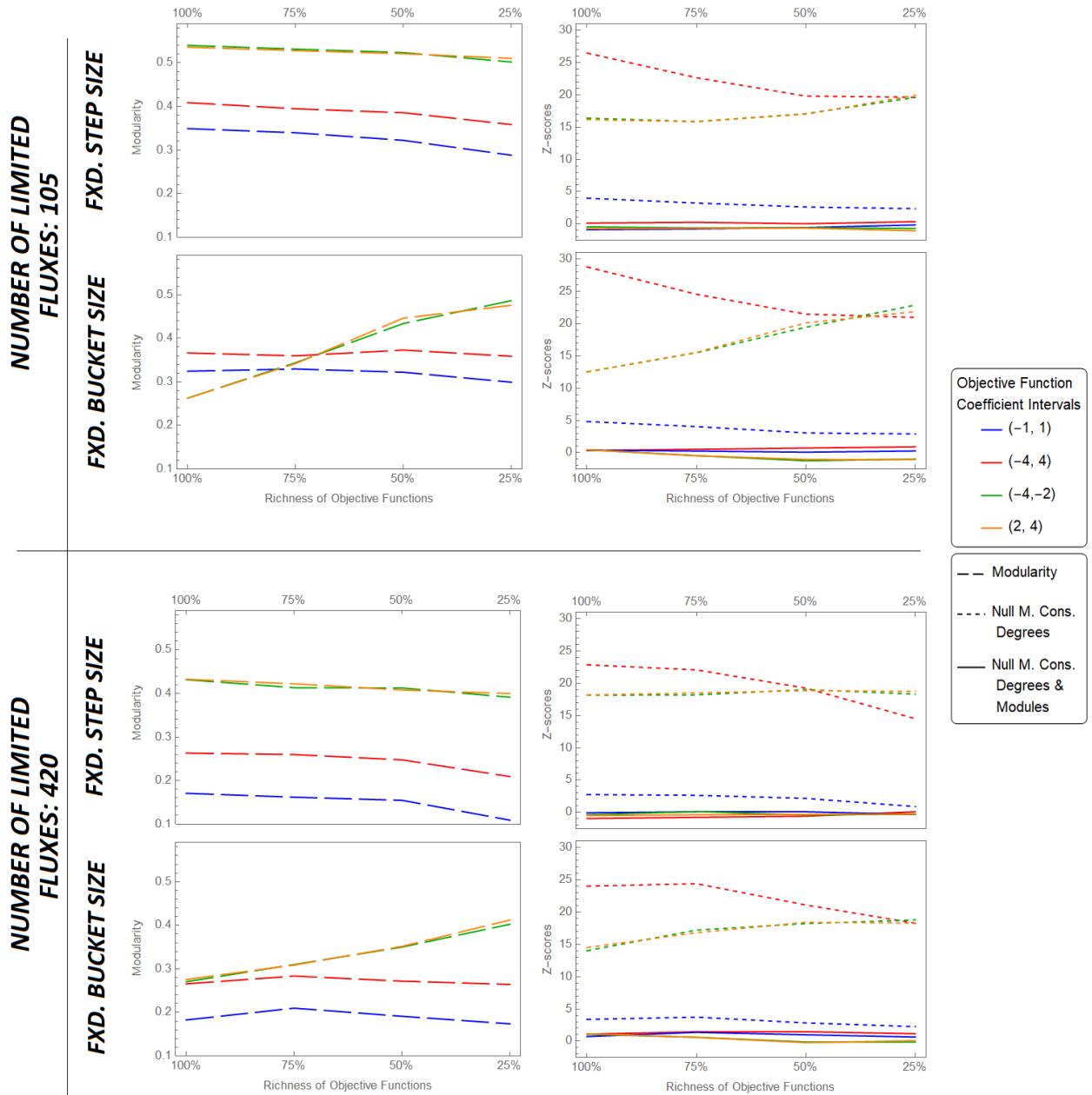


Figure 3.2: Simulation Analysis Curve Plot Results: Modularity Values and Z-scores.

4 Conclusion and Outlook

FBA is a good control model and can be used with a random graph considering some additional consistency constraints. Need to make sure that the cycles in the graph are suitable to create stuff out of nothing. There are some mass balance constraints that need to be incorporated.

Network perturbation might be another further study subject by upgrading the OR-model with advance tools.

5 Bibliography

- [1] J. Waldner and W. Duffin, *CIM : Principles of Computer-integrated Manufacturing*. Wiley, 1992.
- [2] 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.
- [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] SMS group, “Learning steel plant at Big River Steel.” <https://www.sms-group.com/press-media/media/downloads/download-detail/download/15940>. Accessed on 05-07-2021.
- [5] 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.
- [6] 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.
- [7] A.-L. Barabási, *Network Science*. Cambridge University Press, 2016.
- [8] 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.
- [9] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.
- [10] 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.
- [11] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, p. 026113, Feb 2004.

- [12] 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.
- [13] C. Fretter, M. Müller-Hannemann, and M.-T. Hütt, “Subgraph fluctuations in random graphs,” *Phys. Rev. E*, vol. 85, p. 056119, May 2012.
- [14] 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.
- [15] 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.
- [16] E. Ravasz and A.-L. Barabási, “Hierarchical organization in complex networks,” *Phys. Rev. E*, vol. 67, p. 026112, 02 2003.
- [17] 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.
- [18] 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.
- [19] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, *Systems biology in practice: concepts, implementation and application*. John Wiley & Sons, 2005.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.

- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.

List of Figures

1.1 Steel Manufacturing Steps [3].	1
2.1 An Arbitrary Representation for Adjacency Matrix and Its Graph.	8
2.2 Graph Results For Two Different Network Approaches.	9
2.3 Formation of Different Null Models.	11
2.4 Chart Comparing the Various Grading Methods in A Normal Distribution [14].	12
2.5 Network Representations for Homo Sapiens Metabolic Model . .	16
2.6 A Simplified Reaction-centric Network Sketch Shows The Reactions for Exchange, Uptake and Secretion.	18
2.7 Simulation Model Illustration	20
3.1 CCM, CSP, PLTCM, CGL Analysis Bar Chart Results: Z-scores..	26
3.2 Simulation Analysis Curve Plot Results: Modularity Values and Z-scores.	29

List of Tables

1.1 Production Lines Including Various Machine Modules and Compact Solutions	2
2.1 Arbitrary Manufacturing Data Set D	7
2.2 Data Set D with FSS Bin Size Labels.	8
2.3 Data Set D with FBS Bin Size Labels.	9
2.4 Expected Network Structures With Respect to Null Models. . . .	14

List of Equations

2.1 Lift Formula	7
2.2 Modularity Formula	10
2.3 Standard Score Formula	12
2.4 Stoichiometric Matrix	14
2.5 Fluxes Solution Vector	16
2.6 Mass Balance in Steady State	16
2.7 Objective Function Coefficients Array	17
2.8 Maximized Biomass	17
2.9 Constrained Fluxes List	19
2.10 Deleted Fluxes List	19

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: Simulation Results with Initial Terms in Objective Functions.
- S15: Simulation Results with 25% Reduced Terms in Objective Functions.
- S16: Simulation Results with 50% Reduced Terms in Objective Functions.
- S17: Simulation Results with 75% Reduced Terms in Objective Functions.

```

01 |   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
02 |   FROM L3MAIN.r_os ros
03 |   LEFT JOIN L3MAIN.r_ccm ccm ON ros.r_os_id=ccm.r_os_id
04 |   LEFT JOIN L3MAIN.r_ccm_slab slab ON ros.r_os_id=slab.r_os_id
05 |   LEFT JOIN L3MAIN.r_mat mat ON ros.r_os_id=mat.r_os_id
06 |   WHERE sequence_id IS NOT NULL;

```

```

01 |   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
02 |   FROM r_ccm ccm2
03 |   LEFT JOIN (
04 |       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
05 |       FROM r_os ros, r_ccm_slab slab, r_ccm ccm, r_mat mat, r_mat mat2, r_os ros2
06 |       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=(
07 |           SELECT MAX(mat2.modification_date)
08 |           FROM r_mat mat2
09 |           WHERE mat2.material_type='S' AND mat2.material_id=slab.material_id
10 |           ) AND ros.r_os_id=mat.r_os_id AND ros.production_line_name LIKE 'CCM1'
11 |   ) sl ON sl.sequence_id=ccm2.sequence_id
12 |   WHERE ccm2.ladle_arrival_time>to_date('01.07.2017', 'DD.MM.YYYY');

```

Figure S1: SQL Inquiries for CCM and CSP.

```

01 |   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,
02 |   data.thickness_hsm, data.thickness, data.crosssection, data.weight, data.length, data.pickling_temp_avg, data.pickling_speed_avg, data.
03 |   pickling_pressure_avg, data.elongation, data.oiling_flag, data.oil_type, data.operation_mode, data.roll_set_id, data.spm_mode, data.
04 |   yield_point_calc, data.trim_flag, data.trim_width, data.cut_date, data.target_thickness, data.pl_oiling_flag, data.pl_oiling_type, data.
05 |   pl_oiling_weight_top, data.pl_elongation, data.hot_coiling_temp, data.hrc_tensile_str, data.hrc_yield_point, data.input_thickness, data.
06 |   input_width, data.input_length, data.target_width, data.target_length
07 |   FROM pg seq
08 |   LEFT JOIN (
09 |     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.
10 |     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')
11 |     width, NVL(TO_CHAR(mat.hot.thickness), 'null') thickness_hsm, NVL(TO_CHAR(mat.thickness), 'null') thickness, NVL(TO_CHAR(mat.thickness*mat.width
12 |     ), 'null') crosssection, NVL(TO_CHAR(mat.weight), 'null') weight, NVL(TO_CHAR(mat.length), 'null') length, NVL(TO_CHAR(PLTCM.pickling_temp_avg), 'null')
13 |     pickling_temp_avg, NVL(TO_CHAR(PLTCM.pickling_speed_avg), 'null') pickling_speed_avg, NVL(TO_CHAR(PLTCM.pickling_pressure_avg), 'null')
14 |     pickling_pressure_avg, NVL(TO_CHAR(tcm.elongation), 'null') elongation, NVL(TO_CHAR(tcm.oiling_flag), 'null') oiling_flag, NVL(TO_CHAR(tcm.
15 |     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(
16 |     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')
17 |     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')
18 |     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(
19 |     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.
20 |     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')
21 |     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.
22 |     input_length), 'null') input_length, NVL(TO_CHAR(pdi.target_width), 'null') target_width, NVL(TO_CHAR(pdi.target_length), 'null') target_length
23 |     FROM pdi_pltcm pdi, pgl pgl, r_mat mat, r_mat mat_hot, r_PLTCM_IN PLTCM, r_TCM tcm
24 |     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.
25 |     material_id AND mat.material_id=PLTCM.material_id AND mat.material_type LIKE 'CC' AND mat.modification_date=(

26 |       SELECT MAX(modification_date)
27 |       FROM r_mat mat2
28 |       WHERE mat2.material_id=mat.material_id AND mat2.material_type='CC'
29 |     ) AND mat_hot.modification_date=(

30 |       SELECT MAX(modification_date)
31 |       FROM r_mat mat3
32 |       WHERE mat3.material_id=mat.material_id AND mat3.material_type='CH'
33 |     )
34 |   ) data ON data.program_id=seq.program_id
35 |   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.

```

01 |   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
02 |   FROM pg_seq
03 |   LEFT JOIN (
04 |       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
05 |       FROM pdi_cgl pdi, pgl pgl, r_mat mat, r_cgl cgl
06 |       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=(
07 |           SELECT MAX(modification_date)
08 |           FROM r_mat mat2
09 |           WHERE mat2.material_id=mat.material_id
10 |       )
11 |       ) data ON data.program_id=seq.program_id
12 |   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.

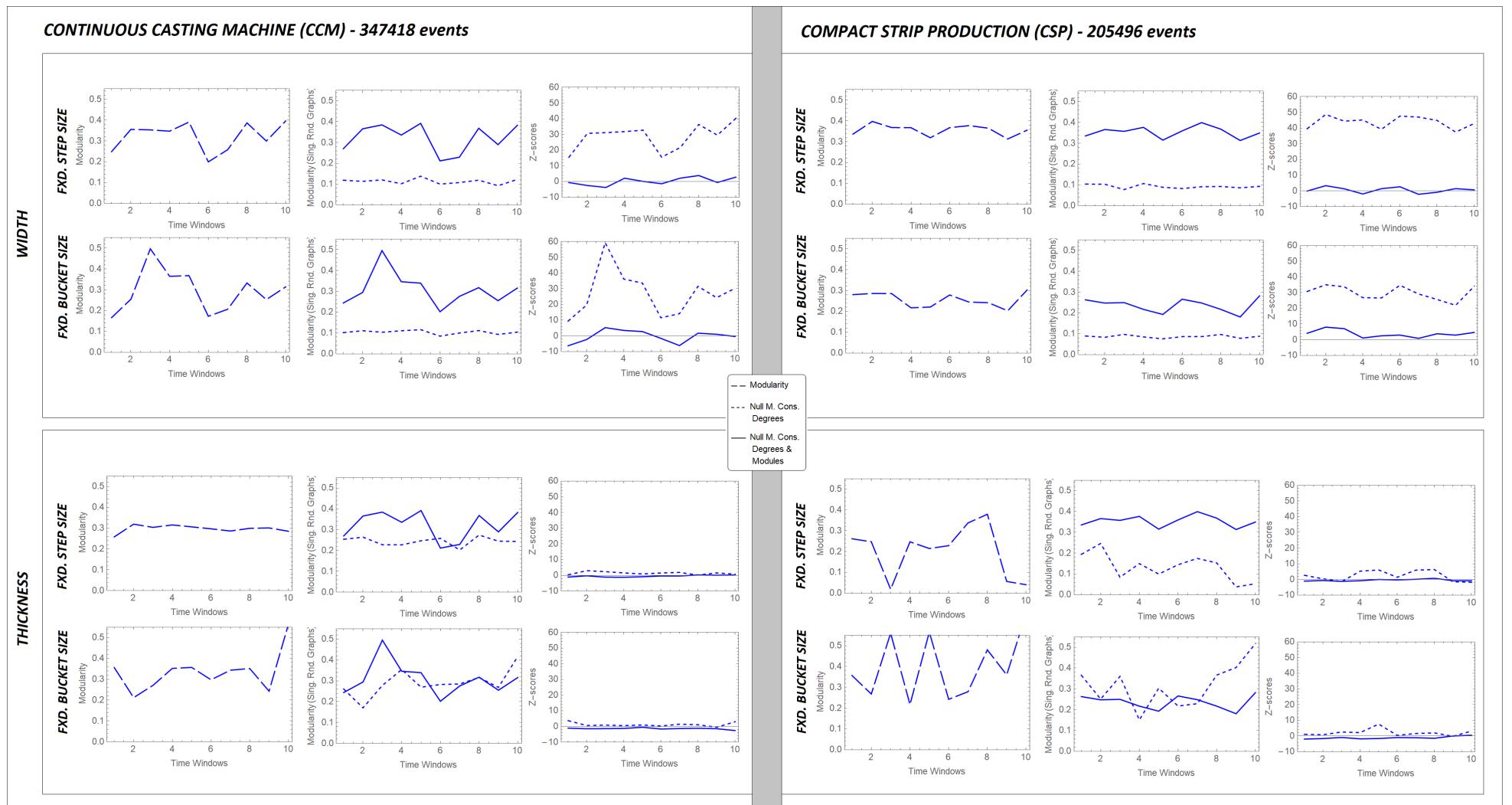


Figure S4: CCM, CSP Analysis Curve Plot Results: Modularity Values and Z-scores in Discrete-time Windows.

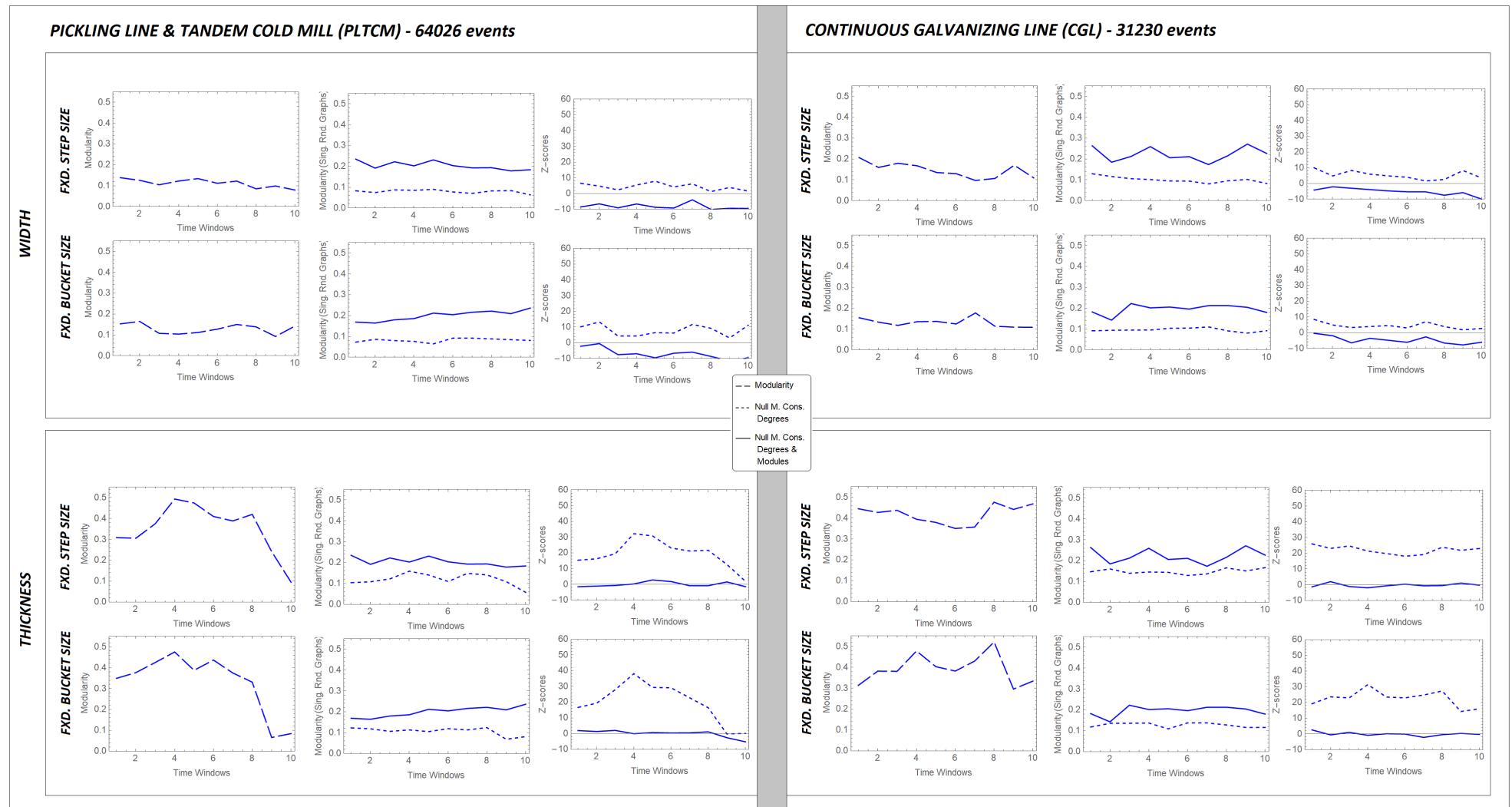


Figure S5: PLTCM, CGL Analysis Curve Plot Results: Modularity Values and Z-scores in Discrete-time Windows.

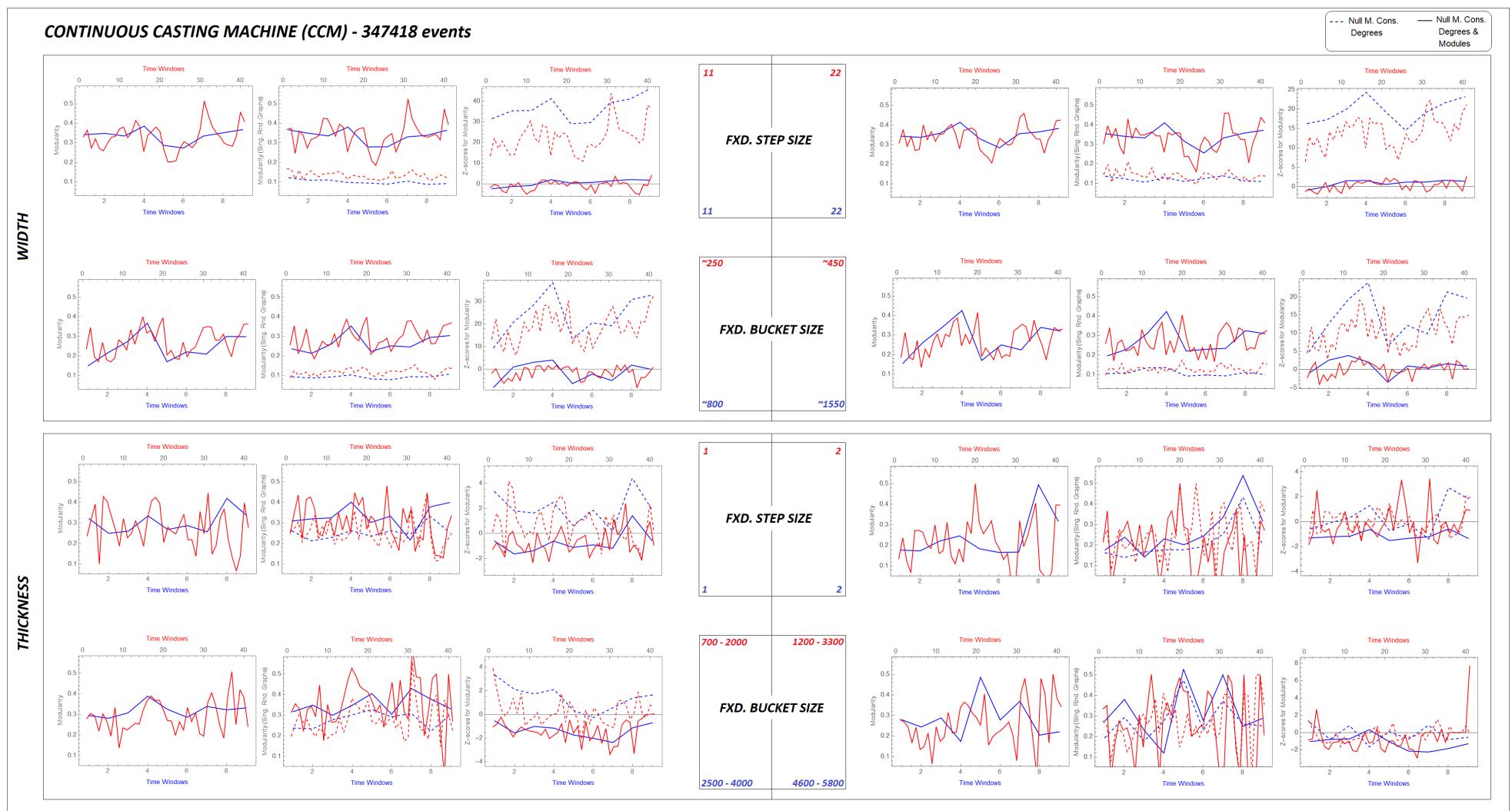


Figure S6: CCM Production Line Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

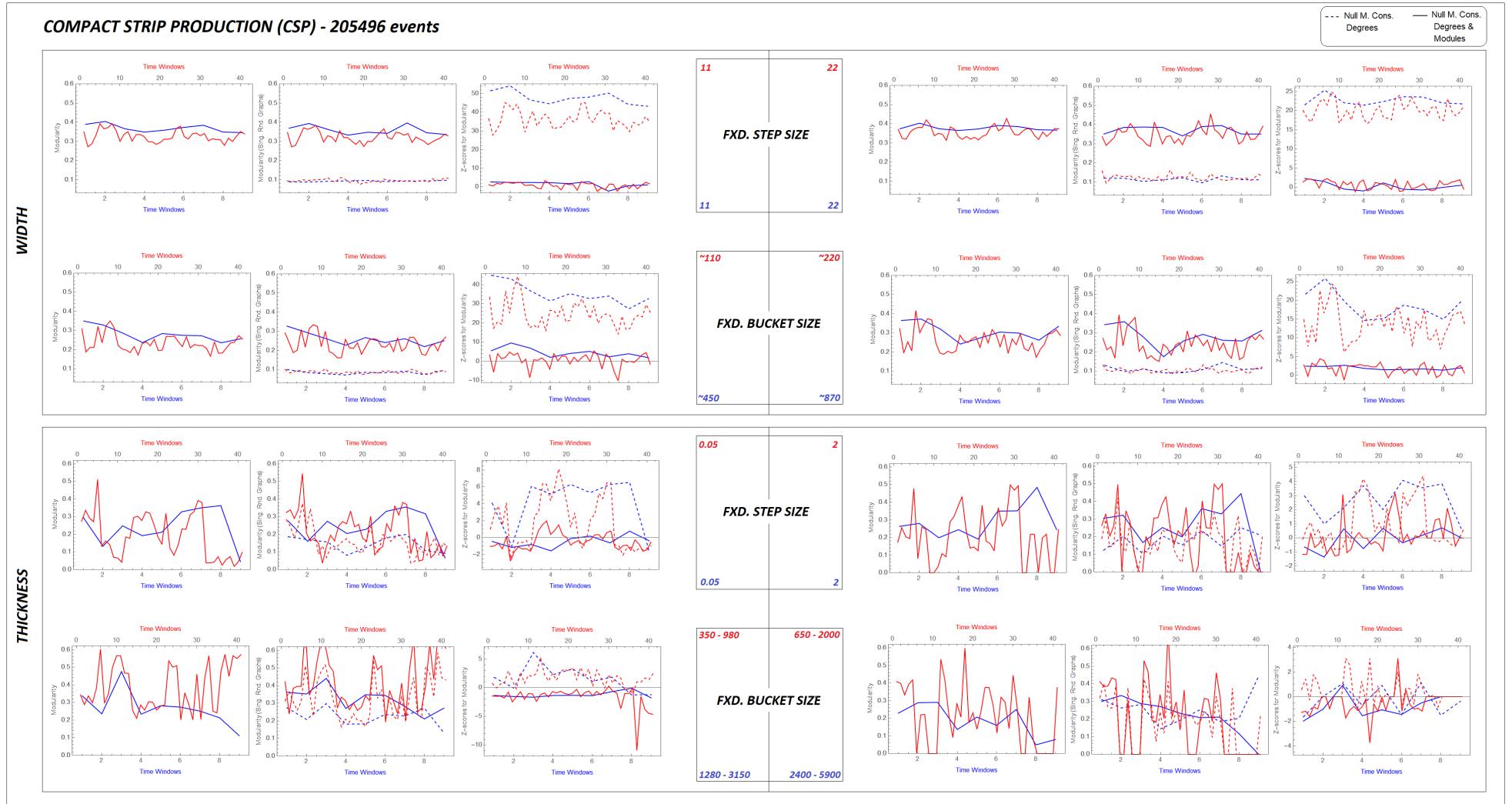


Figure S7: CSP Production Line Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

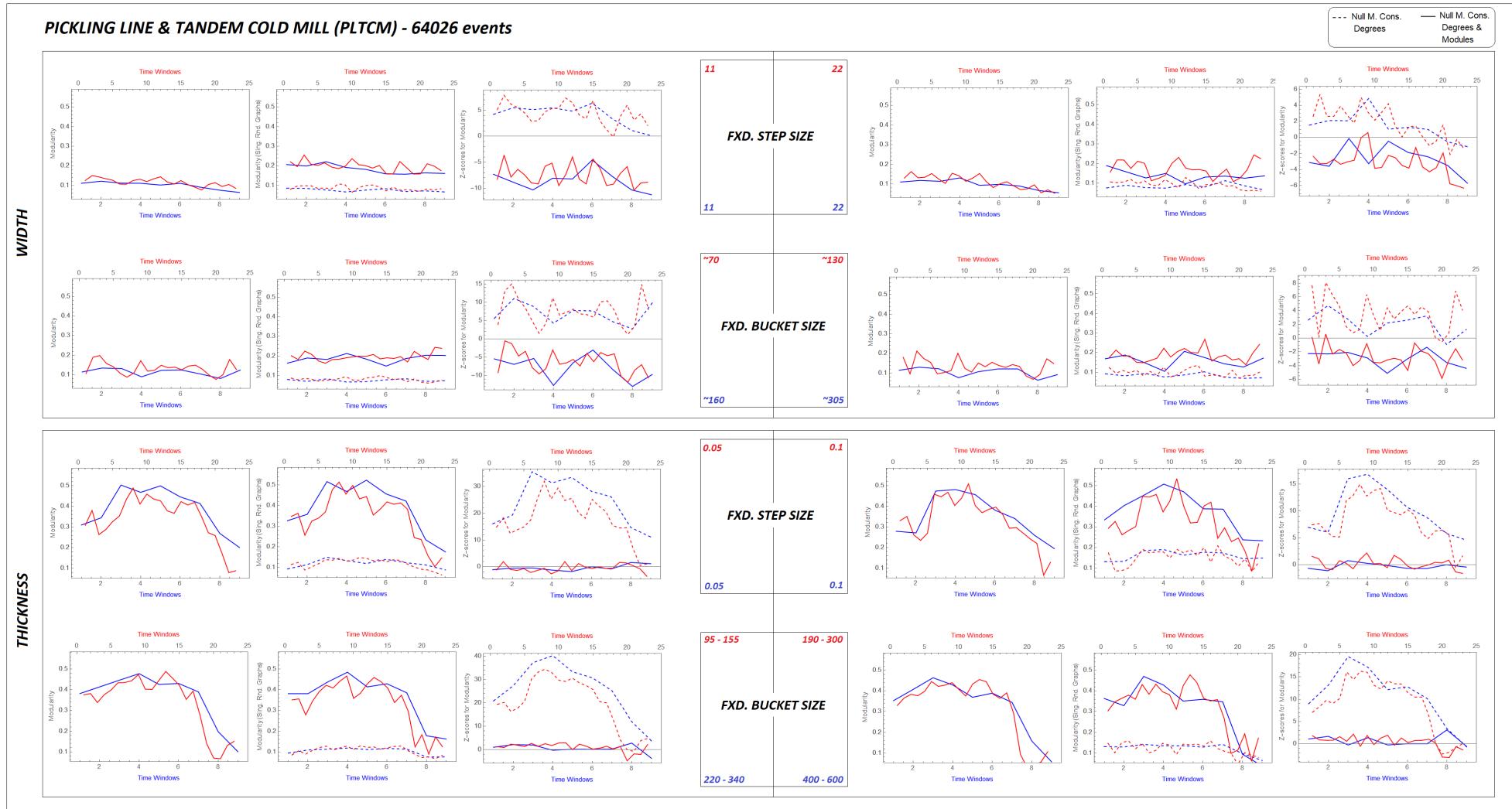


Figure S8: PLTCM Production Line Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

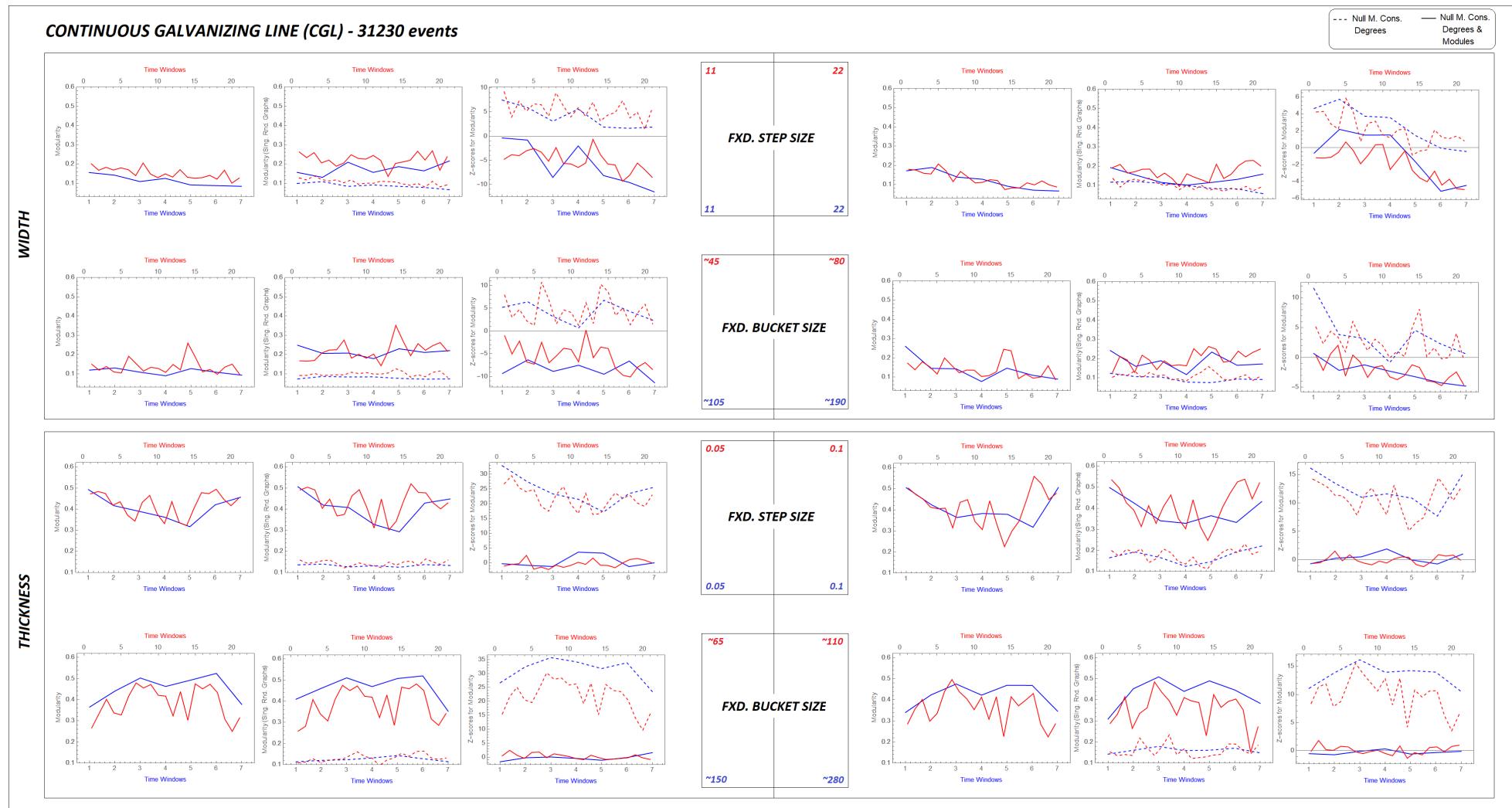


Figure S9: CGL Production Line Analysis Curve Plot Results: Modularity Values and Z-scores in Sliding-time Windows & Different Network Resolutions.

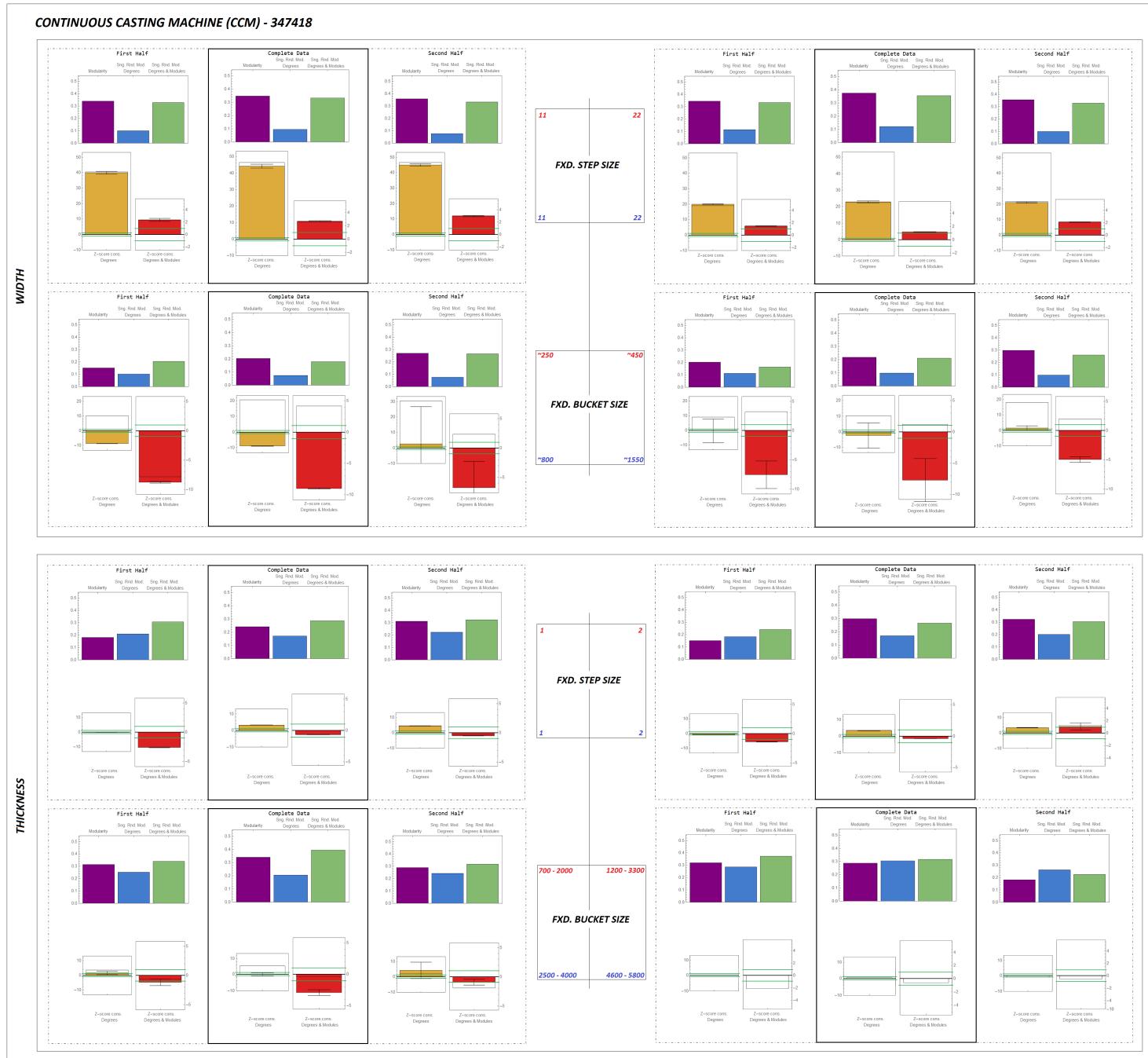


Figure S10: CCM Production Line Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

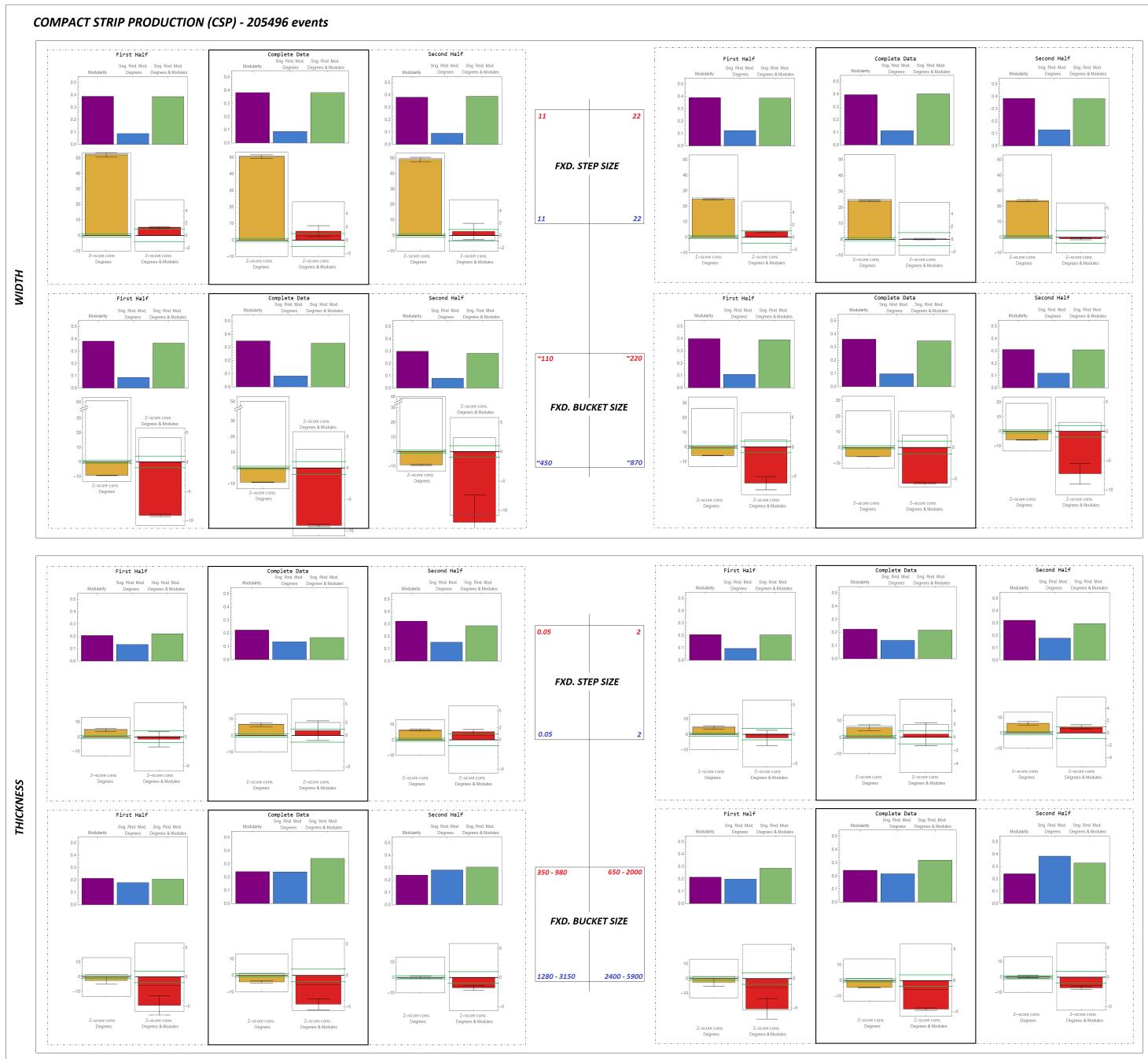


Figure S11: CSP Production Line Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

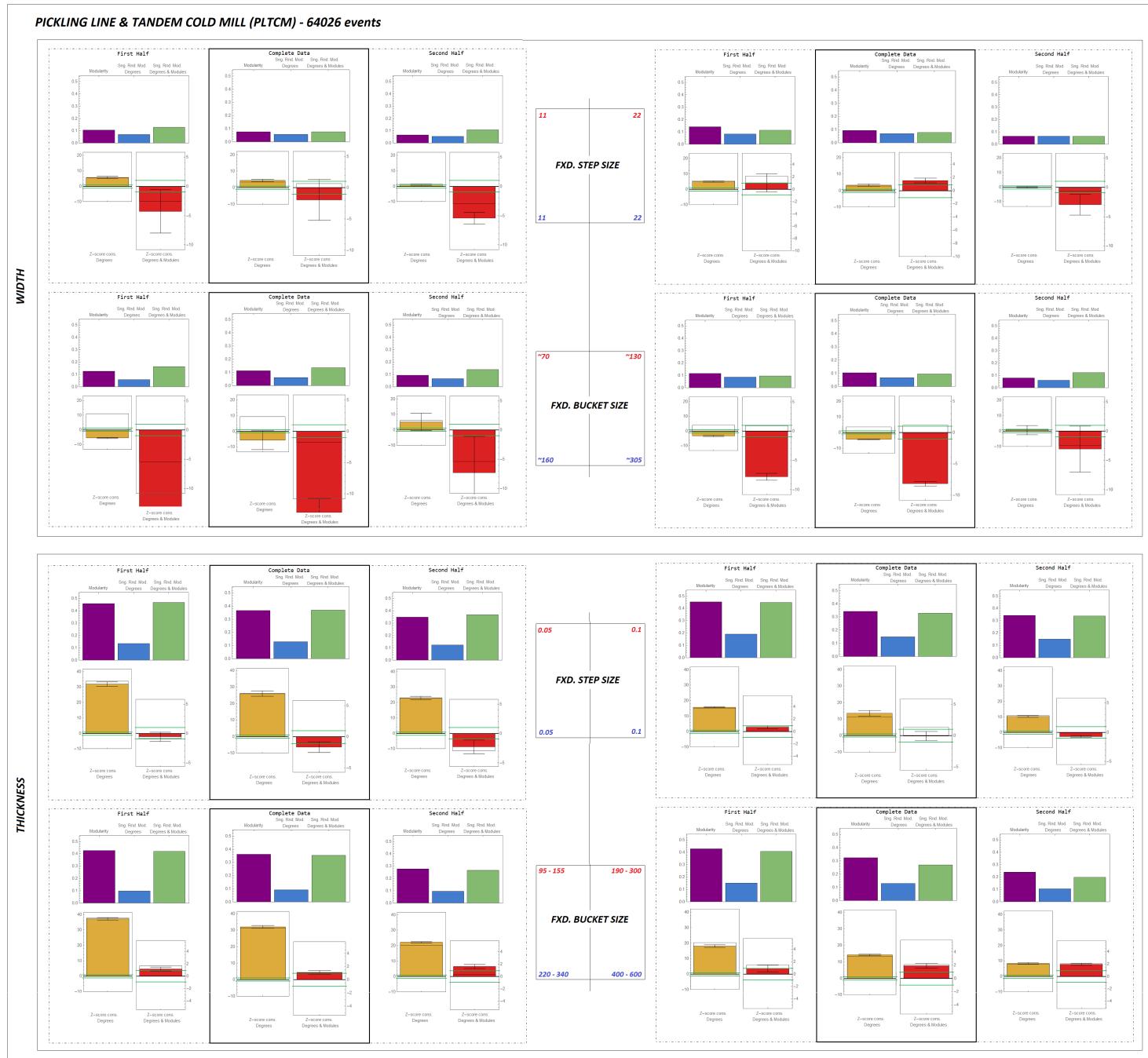


Figure S12: PLTCM Production Line Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

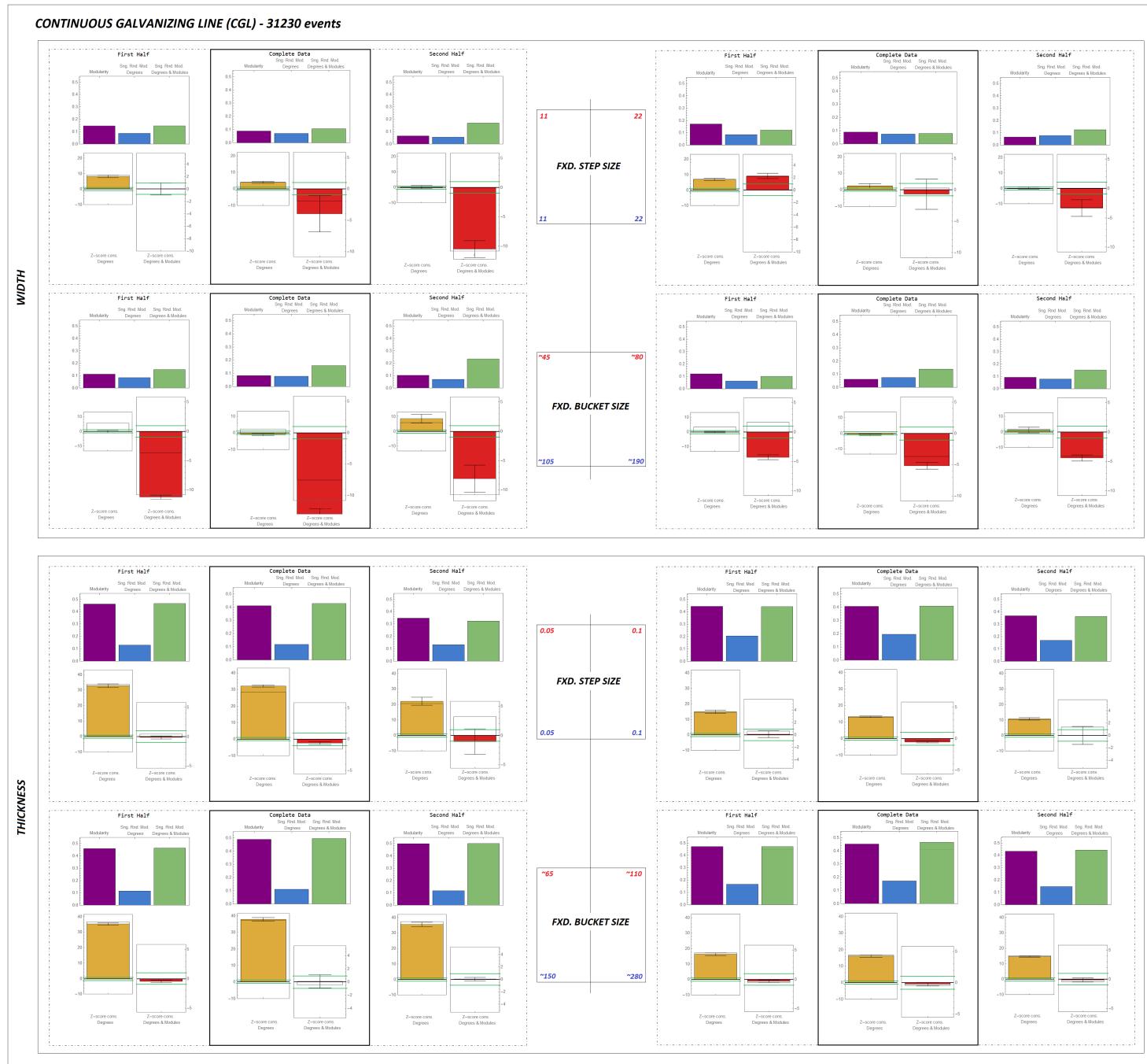


Figure S13: CGL Production Line Analysis Bar Chart Results: Modularity Values and Z-scores in Different Network Resolutions.

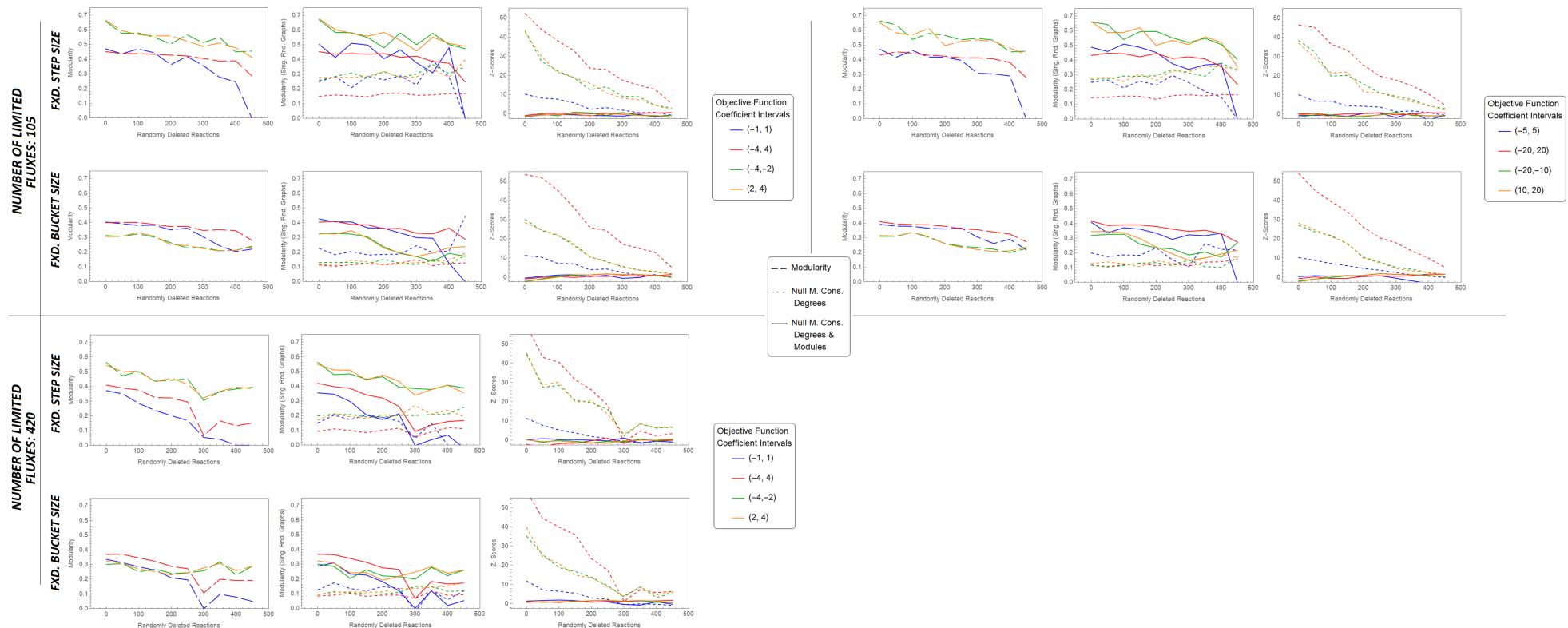


Figure S14: Simulation Results with Initial Terms in Objective Functions.

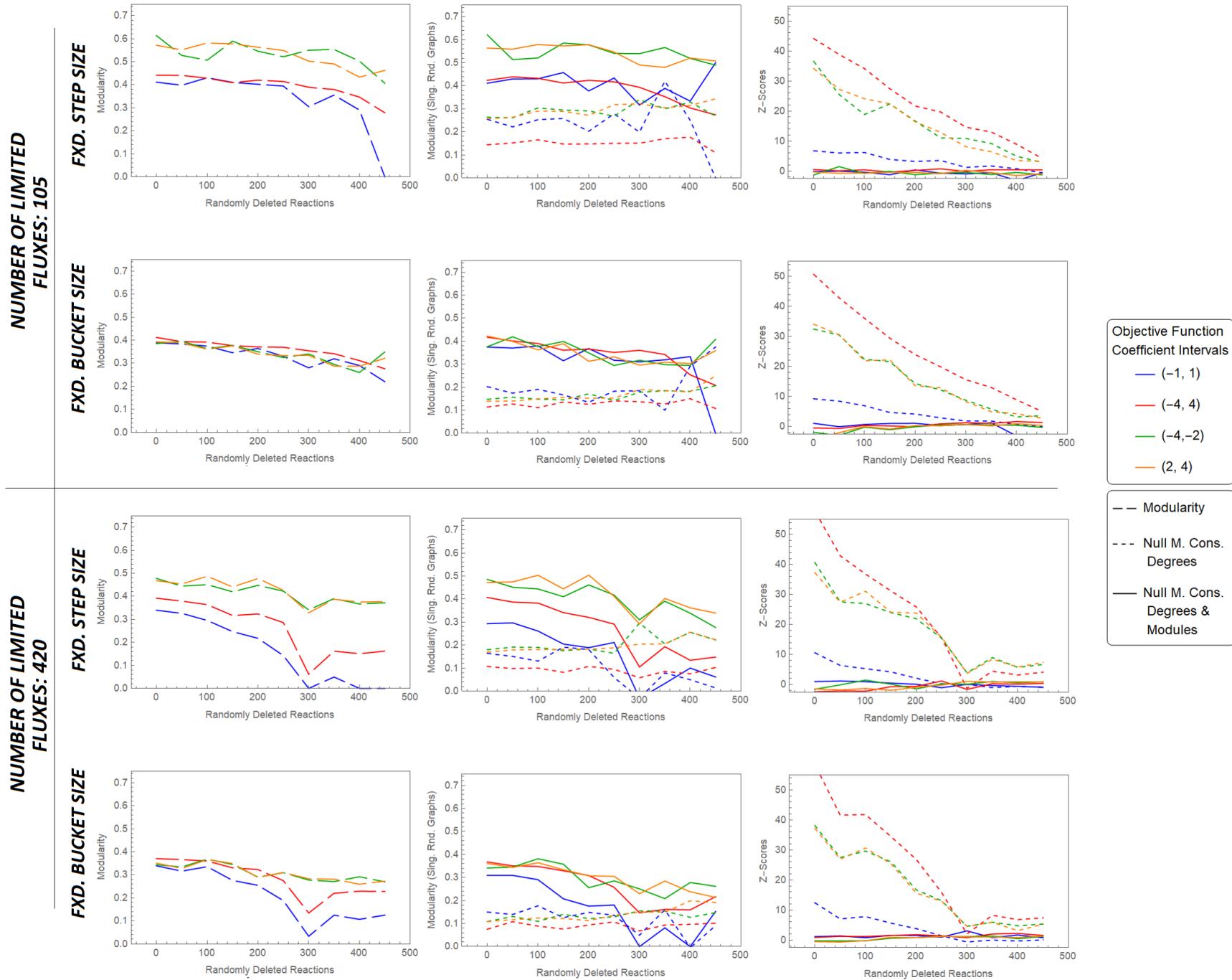


Figure S15: Simulation Results with 25% Reduced Terms in Objective Functions.

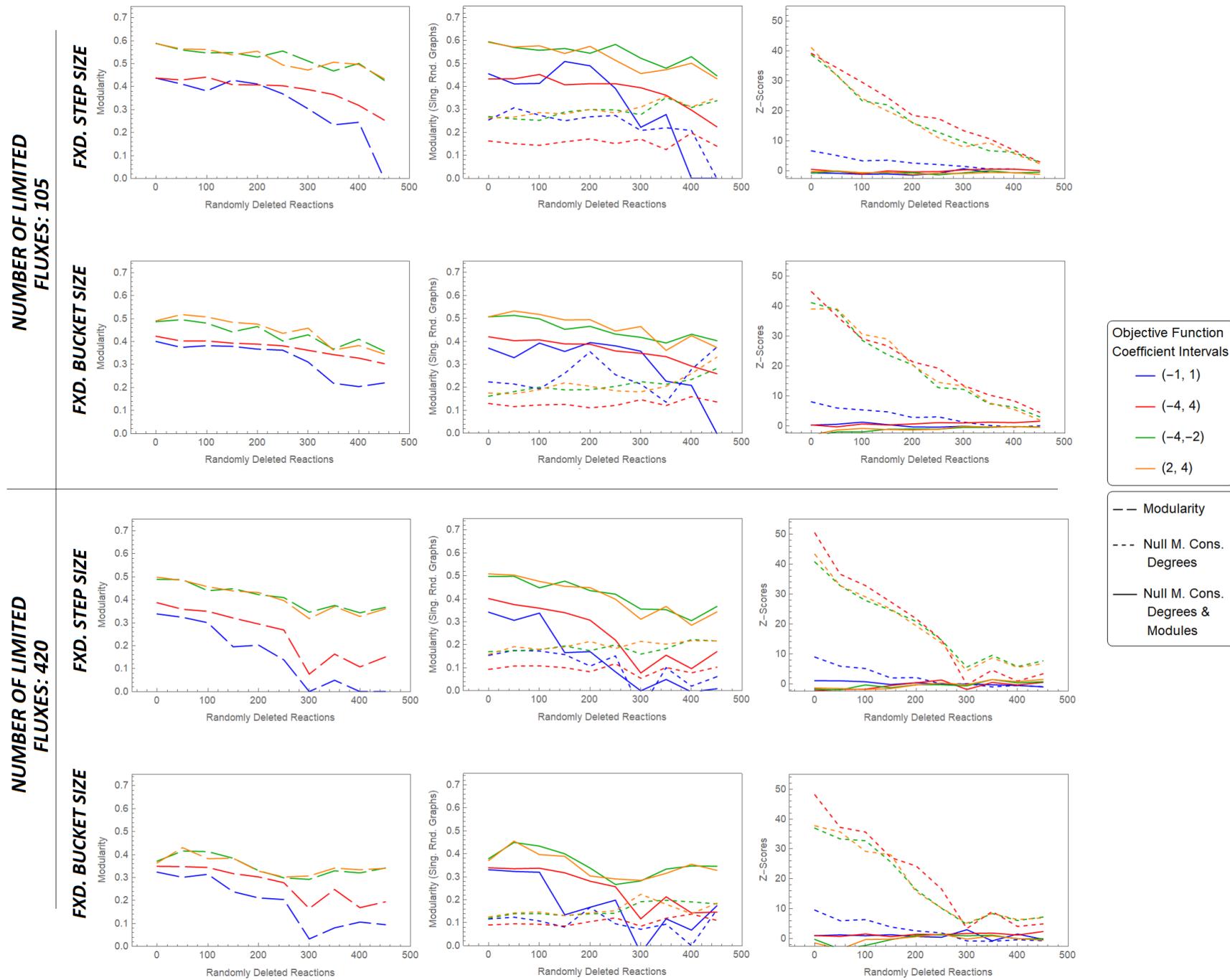


Figure S16: Simulation Results with 50% Reduced Terms in Objective Functions.

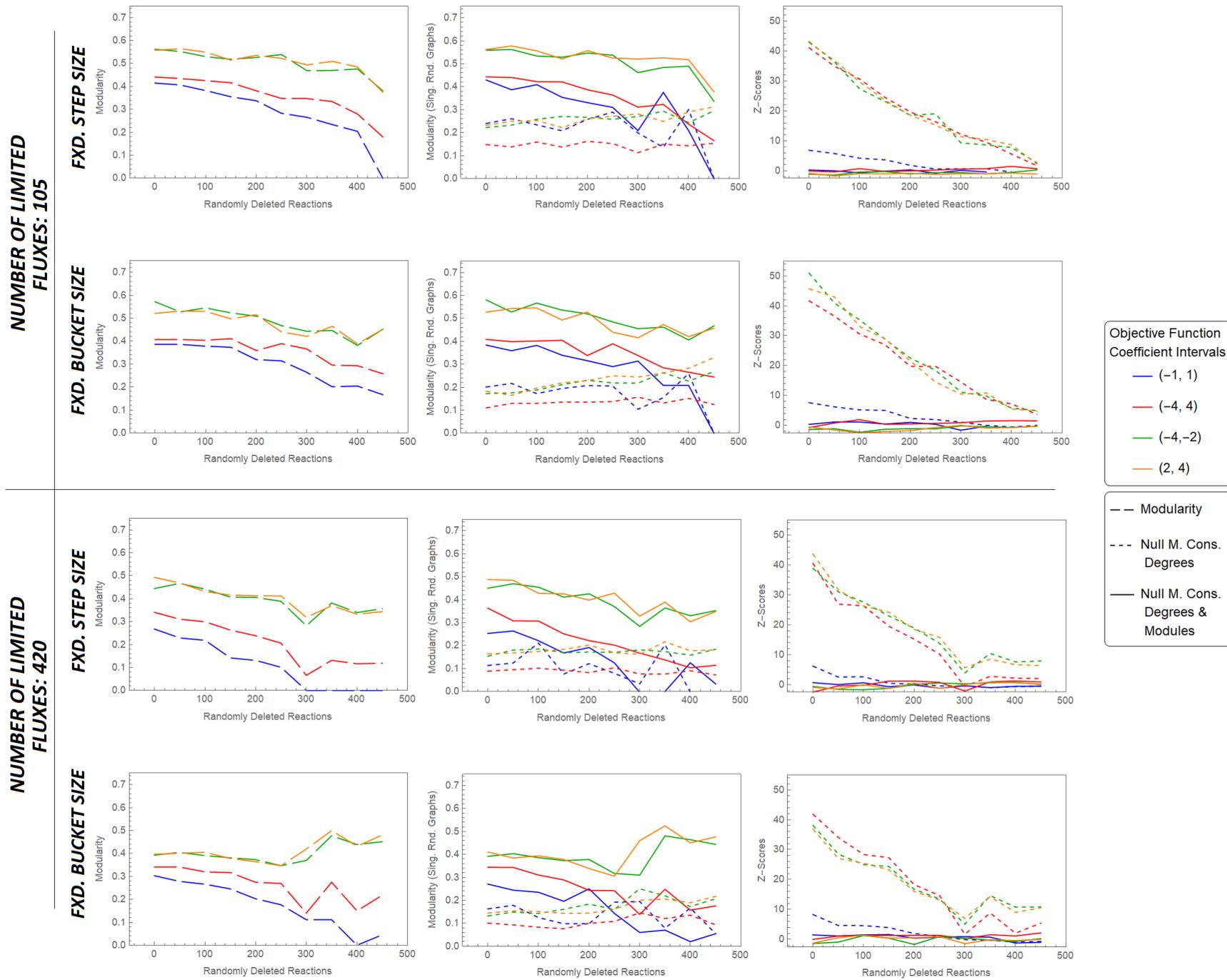


Figure S17: Simulation Results with 75% Reduced Terms in Objective Functions.