An Exploratory Analysis of Errors in and Relationship between Chromatography Database Calculations

DSCI200 August 2024 Project 2 | Ash Baghai

INTRODUCTION

Accurate chemical analyses are essential in pharmaceutical development, and Empower, a leading Chromatography Data System (CDS), plays a crucial role in this process. It is an application used to conduct liquid chromatography analysis of drug compounds, relying on specialized calculations called custom fields. However, as methods evolve, these custom fields can become outdated, and managing them becomes challenging due to the complex interdependencies between fields.

To address this issue, a Python program has been developed that analyzes custom field data from Empower. The program works with a specific tab-delimited, Empower text file export, which contains all relevant custom field information.

The program's two main goals are to detect errors in custom fields and analyze how they relate to one another, by virtue of their references in one another's formulas. By identifying these connections, the program helps streamline the process of correcting, maintaining, and updating custom fields, keeping them in line with current scientific and regulatory standards.

QUESTIONS

1. Can naming and formula errors be detected among custom fields within the Empower Database?
2. How can the impact of custom field (CFs) alterations on other fields be determined?
    a) More specifically, what relationships exist between custom fields based on their references within formulas?

GITHUB REPO

https://github.com/UC-Berkeley-I-School/Project2_Ash_Baghai

DATA SOURCE

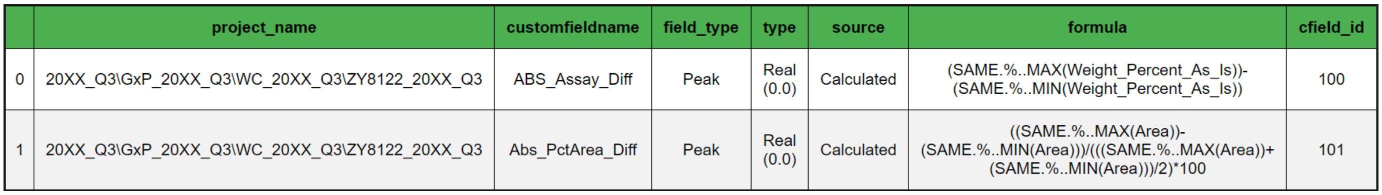
Chromatography Database Data	
Source	The Data file is sourced from an Empower Chromatography Database.
Description	The main dataset is a tab-delimited text file, "CF_Data_File_Project2.txt The data file contains relevant data columns for custom fields associated with experimental chromatography studies.
Primary Data columns of Interest	Column Name: "customfieldname" and "formula" Data type: String Object
Size	336KB 2883 rows, 8 columns

Data Fields	Desicription	Value Example(s)
project_name	The name of a project directory path within Empower where chromatographic methods, custom fields, experimental raw data and processed results would be located.	20XX_Q3\GxP_20XX_Q3\ WC_20XX_Q3\ZY8122_20XX_Q3
customfieldname	The name of particular chromatographic calculations	ABS_Assay_Diff
field_type	The experiment category which the custom field's data relates to	Sample, Sample Set, Peak, or Component
type	The data format of the custom field's output	Real(0.0), Text, or Enum
source	the custom fields output source	Keyboard or Calculated
formula	The equation for a custom field	(SAME.%..MAX(Weight_Percent_As_Is))-(SAME.%..MIN(Weight_Percent_As_Is))

DATA CLEANING

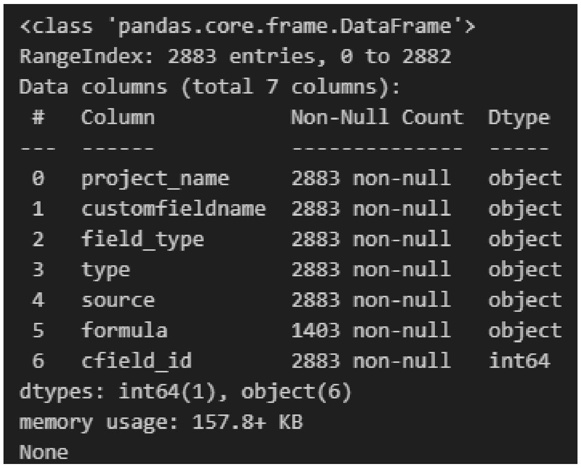
The analysis used a dataset exported from an Empower Database as a tab-delimited text file, serving as the primary input. I imported and formatted the data into a DataFrame using a my own custom program, "eda_formatting," implemented as a Python class. After formatting, I verified that the data columns, fields, and types were accurately represented for further analysis. The data type for the columns of interest, "customfieldname" and "formula" matched expectation, String objects.

Figure A. Confirmation of Data import and formatting



	project_name	customfieldname	field_type	type	source	formula	cfield_id
0	20XX_Q3\GxP_20XX_Q3\WC_20XX_Q3\ZY8122_20XX_Q3	ABS_Assay_Diff	Peak	Real (0.0)	Calculated	(SAME.%..MAX(Weight_Percent_As_Is))-(SAME.%..MIN(Weight_Percent_As_Is))	100
1	20XX_Q3\GxP_20XX_Q3\WC_20XX_Q3\ZY8122_20XX_Q3	Abs_PctArea_Diff	Peak	Real (0.0)	Calculated	((SAME.%..MAX(Area))-(SAME.%..MIN(Area)))/(((SAME.%..MAX(Area))+(SAME.%..MIN(Area)))/2)*100	101

Figure B. Confirmation of Data type



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2883 entries, 0 to 2882
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   project_name     2883 non-null   object
 1   customfieldname  2883 non-null   object
 2   field_type       2883 non-null   object
 3   type             2883 non-null   object
 4   source           2883 non-null   object
 5   formula          1403 non-null   object
 6   cfield_id        2883 non-null   int64
dtypes: int64(1), object(6)
memory usage: 157.8+ KB
None
```

ASSUMPTIONS

Ignore Simultaneous Errors in CF Naming and Formulas

Simultaneous errors in custom field (CF) naming and formulas were not evaluated due to their low probability and the need for a feature outside the current program scope. Detecting such errors would require a ranked similarity search, which could be added in future updates.

Evaluation of CF Relationships Ignores Mathematical Operations

The analysis considered CF relationships based on their occurrence in formulas but did not factor in the associated mathematical operations. Including these operations could enable a weighting function to better assess the impact of each CF on the formula's outcome, a potential improvement for future versions.

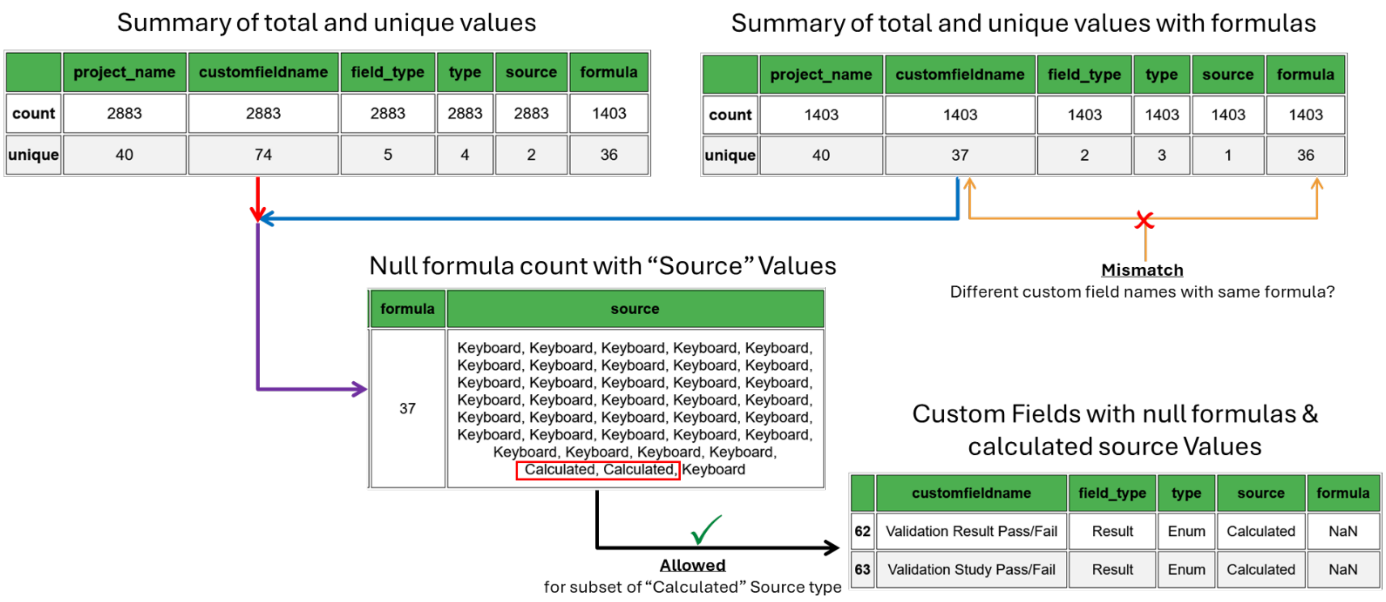
Dropped columns deemed unessential

“#” and “Cfield Id” columns were dropped as they were designated as unessential

ERROR DETECTION ANALYSIS

I began with a basic exploratory data analysis to identify any missing or incorrect values. This process included removing duplicate data since custom fields are often reused across projects, as shown in the top two tables of **Error! Not a valid bookmark self-reference.**.

Figure C. Basic Exploratory Data Analysis Workflow



Summary of total and unique values

	project_name	customfieldname	field_type	type	source	formula
count	2883	2883	2883	2883	2883	1403
unique	40	74	5	4	2	36

Summary of total and unique values with formulas

	project_name	customfieldname	field_type	type	source	formula
count	1403	1403	1403	1403	1403	1403
unique	40	37	2	3	1	36

Null formula count with “Source” Values

formula	source
37	Keyboard, Calculated, Calculated, Keyboard

Custom Fields with null formulas & calculated source Values

	customfieldname	field_type	type	source	formula
62	Validation Result Pass/Fail	Result	Enum	Calculated	NaN
63	Validation Study Pass/Fail	Result	Enum	Calculated	NaN

Handling Empty Formulas

Custom fields relying on manual inputs (“Keyboard” source desingations) are expected to have empty formulas, which is not necessarily the case for those with "Calculated” source designations. For example, programmed custom fields should not have null formulas, unlike default custom fields linked to Empower's built-in calculations. I confirmed that the two identified custom fields with “Calculated” source values were correctly connected to Empower's standard functions, as shown in the bottom section of I began with a basic exploratory data analysis to identify any missing or incorrect values. This process included removing duplicate data since custom fields are often reused across projects, as shown in the top two tables of **Error! Not a valid bookmark self-reference.**.

Figure *C*. This confirmation is by virtue of foreknowlege from conducting Empower administration activites.

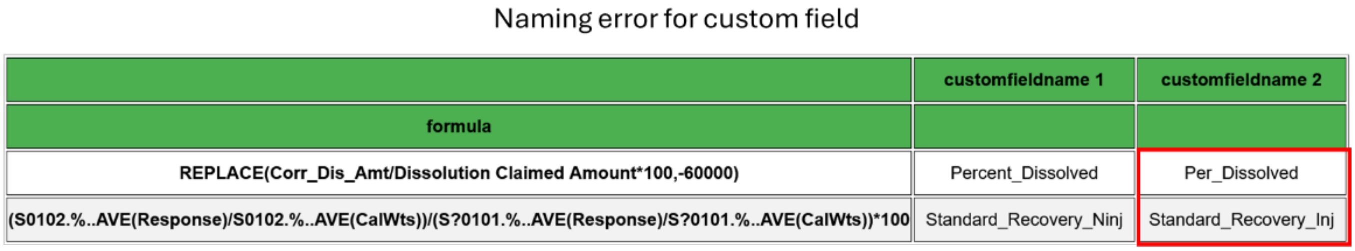
Mismatch Detection in Custom Fields and Formulas
Further analysis revealed a mismatch between the number of unique custom field names and their associated formulas, indicating that a single formula might have multiple name variants. This issue is highlighted in the top right table of I began with a basic exploratory data analysis to identify any missing or incorrect values. This process included removing duplicate data since custom fields are often reused across projects, as shown in the top two tables of **Error! Not a valid bookmark self-reference.**.

Figure *C*.

Identifying and correcting specific error

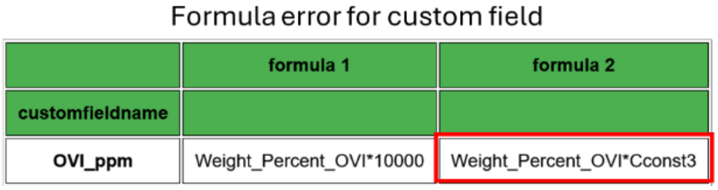
By filtering for custom fields with identical formulas but different names, and vice versa—identical names but different formulas, two naming errors and one formula error were identified; refer to Figure D. In the context of administrative support, these errors would be typically documented and shared with associated Empower administrators for further action. The erroneous data entries were then removed from the DataFrame to ensure accurate results in subsequent analyses.

Figure D. Error Detection



Naming error for custom field

	customfieldname 1	customfieldname 2
formula		
REPLACE(Corr_Dis_Amt/Dissolution Claimed Amount*100,-60000)	Percent_Dissolved	Per_Dissolved
(S0102.%..AVE(Response)/S0102.%..AVE(CalWts))/(S?0101.%..AVE(Response)/S?0101.%..AVE(CalWts))*100	Standard_Recovery_Ninj	Standard_Recovery_Inj



Formula error for custom field

	formula 1	formula 2
customfieldname		
OVI_ppm	Weight_Percent_OVI*10000	Weight_Percent_OVI*Cconst3

Detected errors are boxed in red.

IDENTIFYING CUSTOM FIELD RELATIONSHIPS

Recursive Analysis

A recursive analysis was conducted to assess how custom fields depend on one another within the dataset. This analysis aimed to highlight the disproportionate influence certain custom fields have and how errors in these fields could propagate throughout the system.

Table 1 summarizes the direct and nested references of custom fields in other custom field formulas, either directly or indirectly through nested references. The bar graph below in

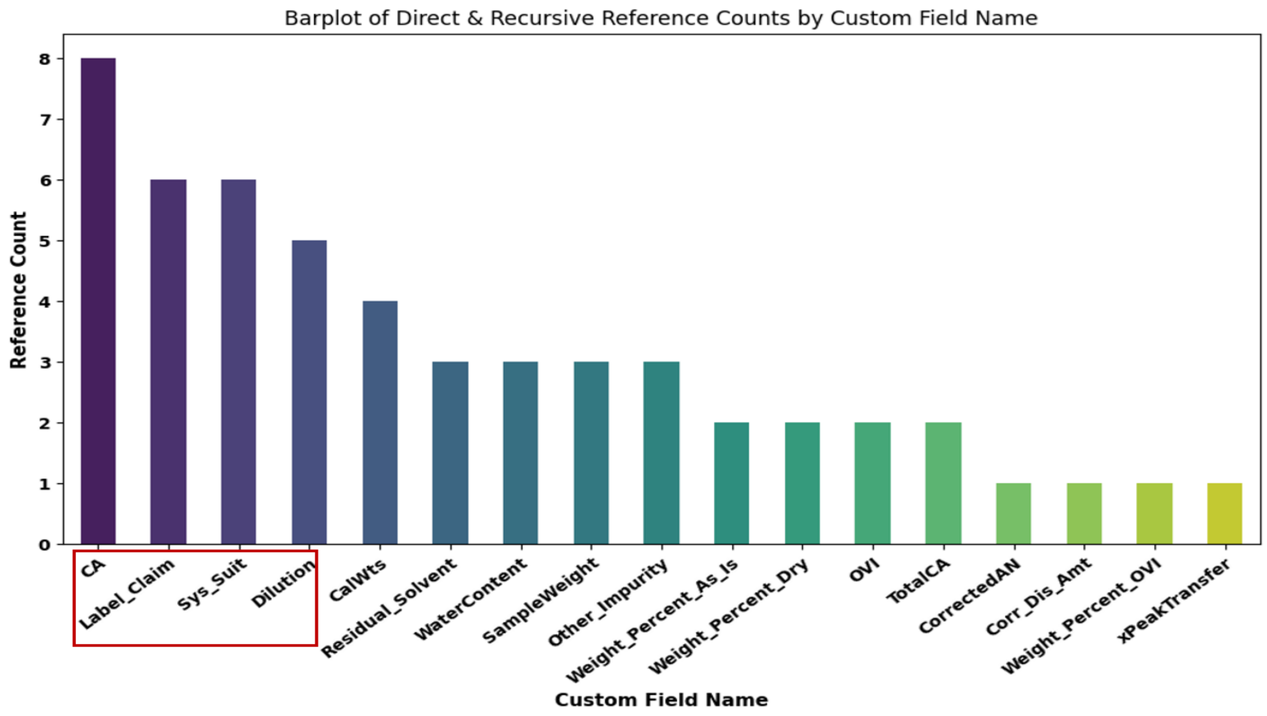
Figure *E*, shows the frequency of these references across all formulas. Notably, custom fields like CA (Corrected Area), Label_Claim, Sys_Suit (System Suitability), and Dilution have the highest reference counts, indicating their significant role in the dataset and the potential impact of any errors associated with them.

Table 1. Listing of Unique custom field referenced in other custom field formulas

Custom Field Name	**CF References**
ABS_Assay_Diff	['Label_Claim', 'Weight_Percent_As_Is']
AR_Sensitivity	['Dilution']
AVE_CorrectedAN	['CA', 'TotalCA', 'CorrectedAN']
CalWts	['Dilution', 'SampleWeight']
CorrectedAN	['TotalCA', 'CA']
LIMS_TRANSFER_ATTRIBUTE	['xPeakTransfer']
OVI	['Other_Impurity', 'Residual_Solvent']
OVI_ppm	['Weight_Percent_OVI']
Percent_Assay_Difference	['Label_Claim', 'Other_Impurity', 'WaterContent', 'OVI', 'Residual_Solvent', 'Weight_Percent_Dry']
Percent_Dissolved	['Corr_Dis_Amt']
R_USP_Plates	['Sys_Suit']
R_USP_Resolution	['Sys_Suit']
R_USP_Tailing	['Sys_Suit']
Sensitivity	['Dilution']
Standard_Recovery	['CalWts', 'SampleWeight', 'Dilution']
Standard_Recovery_Ninj	['CalWts', 'SampleWeight', 'Dilution']
TotalAreaGroup	['CA']
TotalCA	['CA']
Weight_Percent_Anhydrous	['Label_Claim', 'WaterContent']
Weight_Percent_As_Is	['Label_Claim']
Weight_Percent_Dry	['Label_Claim', 'Other_Impurity', 'WaterContent', 'OVI', 'Residual_Solvent']
xSampleWeight	['Label_Claim']

Custom fields are highlighted based on frequency ranking in Figure F

Figure E. The frequency count for direct and recursively referenced custom fields



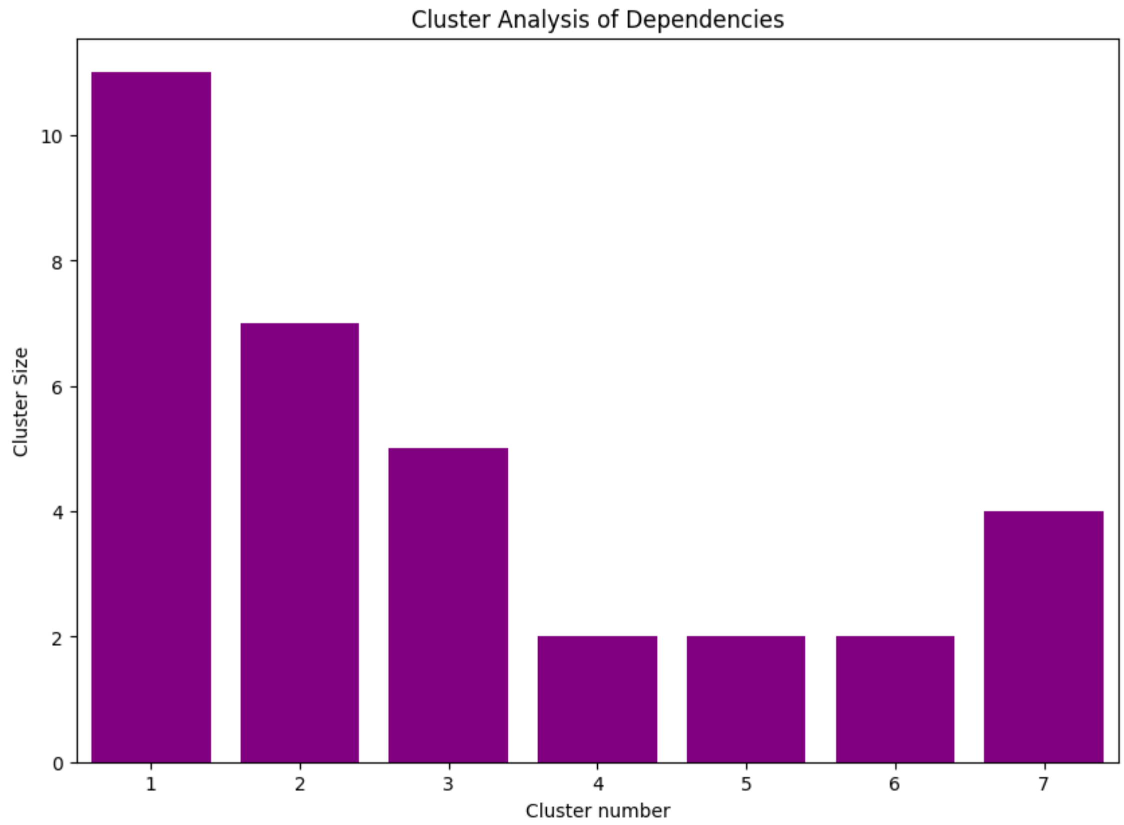
Cluster Analysis

A cluster analysis, using the python NetworkX library, was conducted to identify natural groupings of custom fields based on formula similarities and weakly connected components; this signifies direct and intermediate connections. This analysis provides some added context to what was identified in the recursive analysis by revealing how the custom fields may be associated. These insights could offer guidance on which custom fields to consider for similar updates based on their formula similarities.

Table 2. Cluster Analysis: number of Connected Custom Fields (Nodes) with a given cluster

Clusters	# Nodes	CF Dependencies
Cluster 1	11	Label_Claim, Residual_Solvent, Weight_Percent_Anhydrous, Other_Impurity, xSampleWeight, WaterContent, ABS_Assay_Diff, OVI, Percent_Assay_Difference, Weight_Percent_Dry, Weight_Percent_As_Is
Cluster 2	7	AR_Sensitivity, CalWts, SampleWeight, Standard_Recovery, Sensitivity, Dilution, Standard_Recovery_Ninj
Cluster 3	5	AVE_CorrectedAN, CA, TotalCA, CorrectedAN, TotalAreaGroup
Cluster 4	2	LIMS_TRANSFER_ATTRIBUTE, xPeakTransfer
Cluster 5	2	Weight_Percent_OVI, OVI_ppm
Cluster 6	2	Corr_Dis_Amt, Percent_Dissolved
Cluster 7	4	R_USP_Tailing, R_USP_Plates, R_USP_Resolution, Sys_Suit

Figure F. Barplot of Cluster Analysis: Cluster Number vs Cluster Size

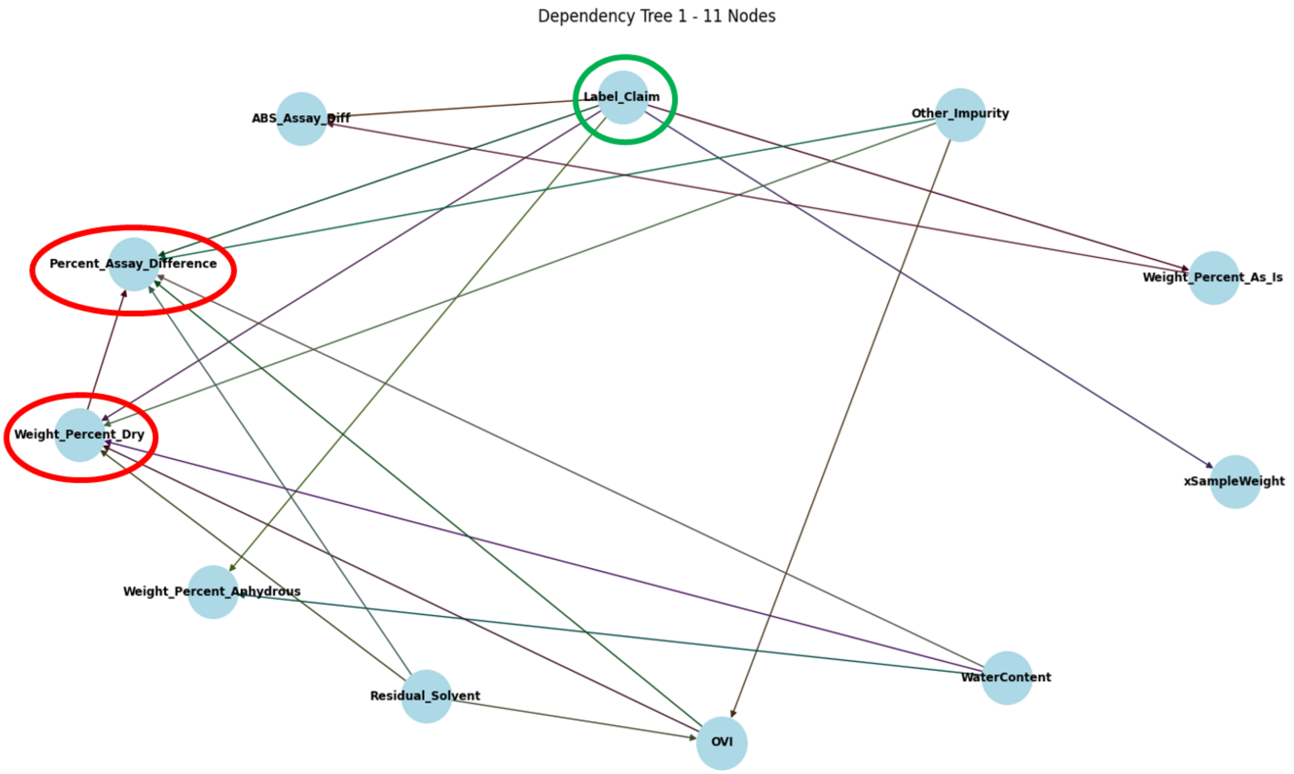


Network Graph Analysis

A directed graph was then used to visualize network relationships within each cluster by mapping the direct and intermediate connections between nodes.

Figure G illustrates the relationships between custom fields within cluster 1, where arrows point from one custom field to another that depends on it. Here, the custom field, "Label_Claim" plays a central role, being referenced by many other custom fields, as indicated in green. In contrast, the custom fields, "Percent_Assay_Difference" and "Weight_Percent_Dry" are highly reliant on multiple sources for their calculations as indicated in red. This visualization helps identify key custom fields that influence or are influenced by others.

Figure G. Custom field, Network Dependency Tree for Cluster 1



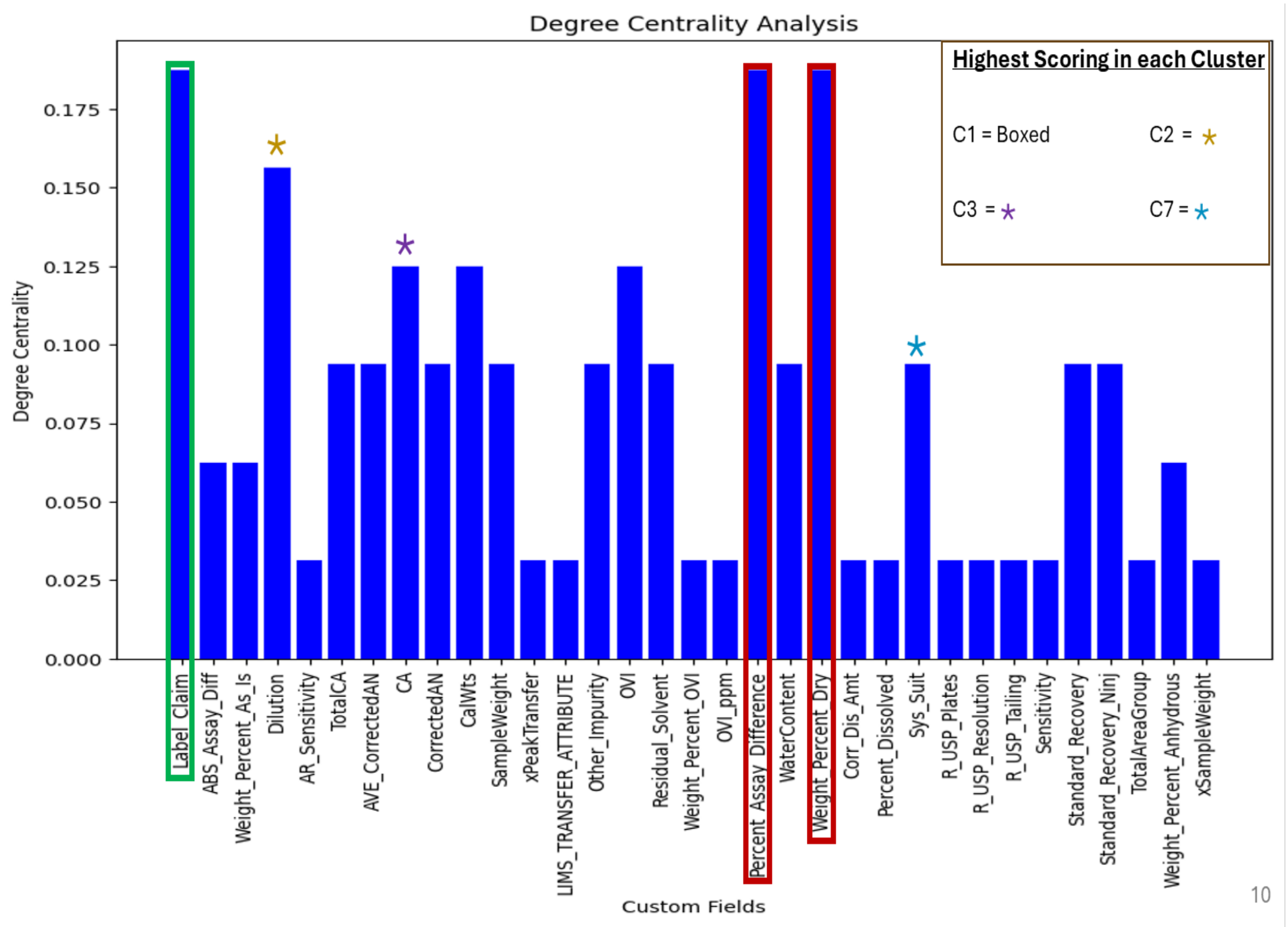
Note: Network Graph Analyses were carried out on all network clusters with greater than 2 nodes. Only one graph is depicted here as it conveys the central insight from the analyses. The remainders are included in the appendices.

Network Centrality Analysis

Finally, a network centrality analysis was conducted to quantify the influence of custom fields within the network. The analysis measures the degree centrality, which indicates how much a node (custom field) within a cluster influences or is influenced by other nodes. The graph from Figure H shows that "Label_Claim," "Percent_Assay_Difference," and "Weight_Percent_Dry" have relatively high degree centrality values, confirming their significant roles as visualized in the Cluster 1 dependency tree.

Additionally, custom fields central to the other clusters of interest are marked with asterisks, highlighting their importance within their respective clusters.

Figure H. Network Centrality Analysis: Custom Fields vs Degree Centrality



CONCLUSION

The exploratory data analysis (EDA) successfully identified three key errors within the dataset: two naming errors and one formula error.

A recursive analysis highlighted the significant reliance on four top custom fields, with the degree of dependence ranked as follows: CA > Label_Claim = Sys_Suit > Dilution.

The cluster analysis revealed seven distinct clusters of linked custom fields. These clusters offer valuable insights for future updates, allowing for more targeted modifications within specific categories.

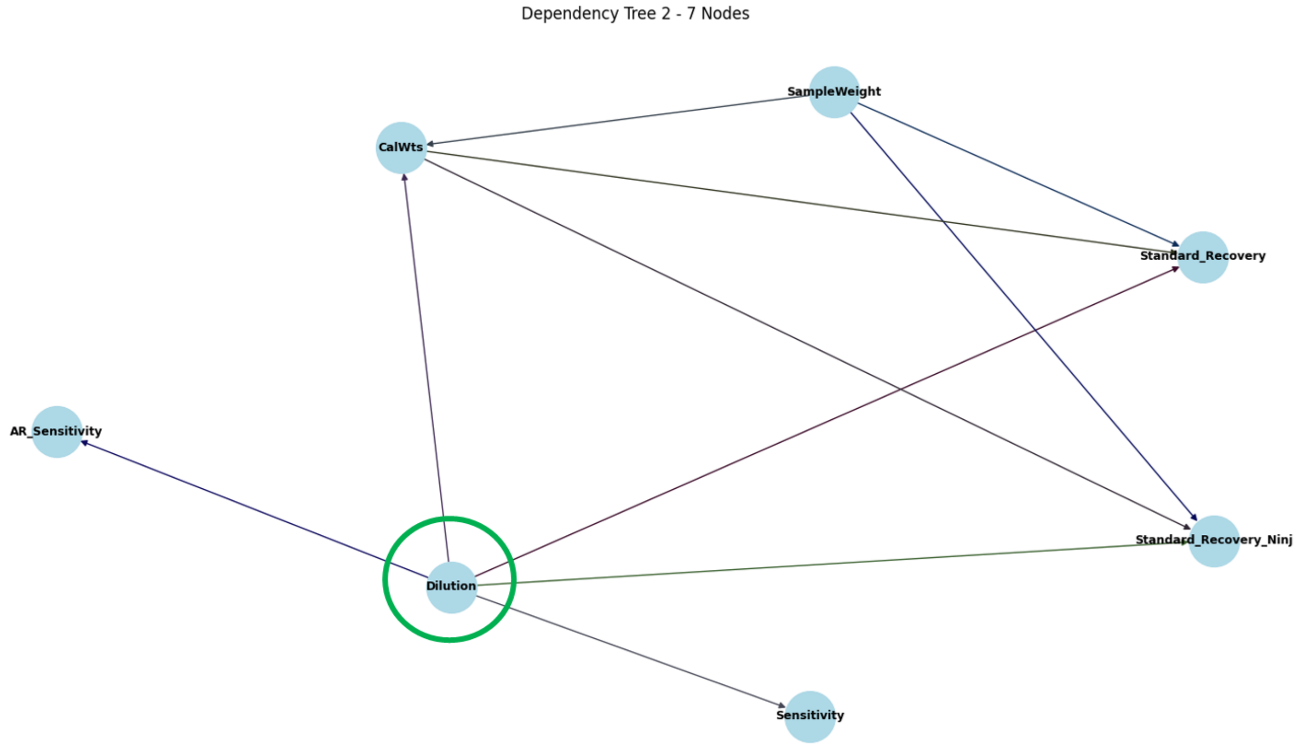
Finally, the network dependency tree and degree centrality analysis identified influential custom fields within each cluster. Notably:

Cluster #	Custom Field(s)
1	Label_Claim, Percent Assay Difference, and Weight_Percent_Dry.
2	Dilution
3	CA
7	Sys_Suit

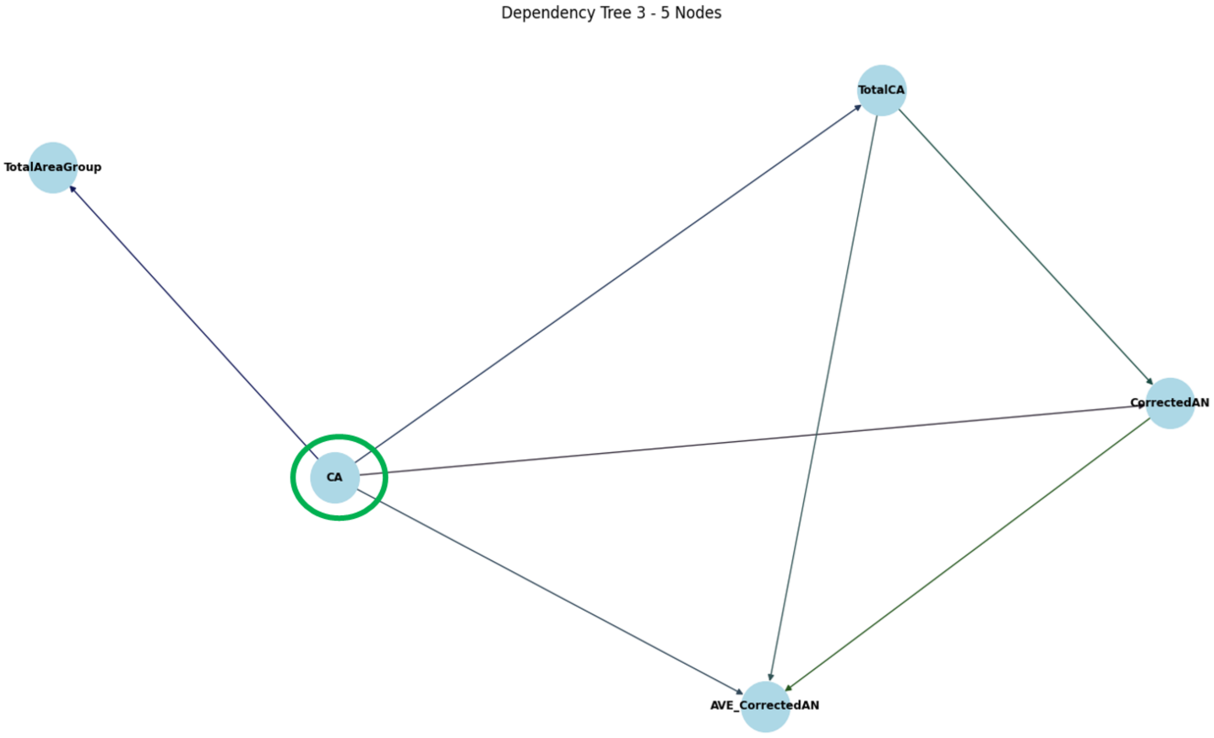
These findings provide a robust framework for error correction and understanding the complex interdependencies of custom fields to guide effective Empower database management and updates.

APPENDIX

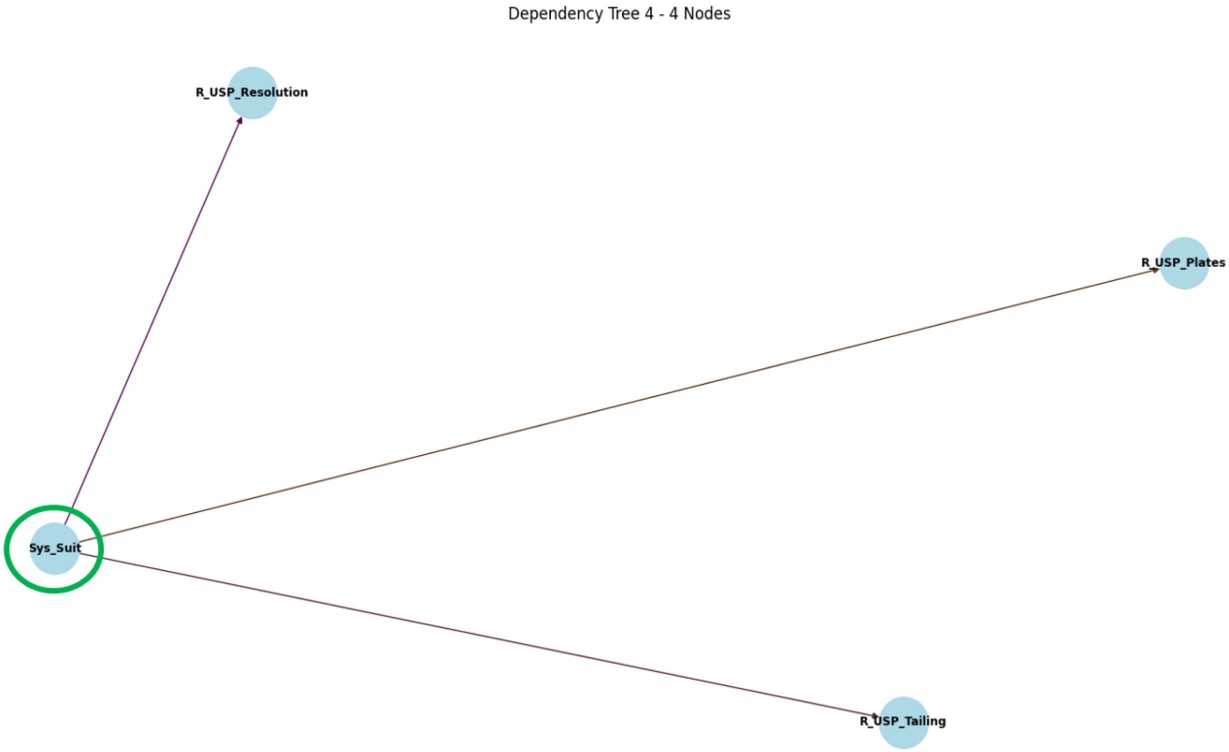
Figurere A1. Custom field, Network Dependency Tree for Cluster 2



Figurere A2. Custom field, Network Dependency Tree for Cluster 3



Figurere A3. Custom field, Network Dependency Tree for Cluster 7


Dependency Tree 4 - 4 Nodes
R_USP_Resolution
R_USP_Plates
Sys_Suit
R_USP_Tailing