DigWise Technology Corporation, LTD.

Copernic™ User Guide

Table of Contents

1. Introduction

The **Copernic™** offers a comprehensive solution for Design-Technology Co-optimization (DTCO), leveraging data analysis and machine learning to effectively enhance chip yield, energy efficiency, and reliability. The system features advanced data visualization, high-dimensional plotting, feature correlation analysis, data modeling, dynamic cross-probing, and powerful APIs. By connecting high-dimensional data, users can quickly gain insights into parameter relationships, understand trends, and scientifically design margins, effectively minimizing the duration of process recipe optimization.

2. Core Features

- **Consolidated Database:** Facilitates interactive querying.
- **Intuitive Graphical User Interface:** Ensures easy navigation and visualization.
- **Batch Execution:** Supports comparisons and trend analyses across multiple wafers.
- **Powerful APIs:** Provide users with advanced data analysis and tool development capabilities.

3. Data Format

To ensure seamless integration and analysis within the Copernic™, it is essential to understand the required data format for Chip Probing (CP) and Wafer Acceptance Test (WAT) data. The following sections outline the expected structures for data type:

- **File Type**: CSV or XLSX or PICKLE.
- **Required Columns**:
  - **LWID**: A distinctive identifier assigned to each wafer for tracking and analysis. (Lot Id. + Wafer No.)
  - **X**: Represents the horizontal coordinate, uniquely identifying the chip's position on the wafer.
  - **Y**: Represents the vertical coordinate, uniquely identifying the chip's position on the wafer.
  - **Features**: Measurement parameters (refer to TABLE I and Fig. 1).

TABLE I: The wafer features in the dataset.

Feature	Description	Unit
CP1	Leakage current	μA
CP2	Chip speed	Hz
CP3	Functional accuracy at 300MHz	%
CP4	Functional accuracy at 400MHz	%
CP5	Functional accuracy at 500MHz	%
CP6	Functional accuracy at 600MHz	%
WAT1	Gate threshold voltage of the low threshold NMOS	V
WAT2	Gate threshold voltage of the low threshold PMOS	V
WAT3	Gate threshold voltage of the ultra-low threshold NMOS	V
WAT4	Gate threshold voltage of the ultra-low threshold PMOS	V
WAT5	Drain current of the low threshold NMOS	mA
WAT6	Drain current of the low threshold PMOS	mA
WAT7	Drain current of the ultra-low threshold NMOS	mA
WAT8	Drain current of the ultra-low threshold PMOS	mA

LWID	X	Y	CP1	CP2	CP3	CP4	CP5	CP6	WAT1	WAT2	WAT3	WAT4	WAT5	WAT6	WAT7	WAT8
1	30	1	0.050494	7.24931	1	1	1	1	0.09654	0.149651	0.079528	0.095597	7.75271	2.77679	8.99698	6.12671
1	31	1	0.214836	70.6877	1.0784	1	1	1	0.101933	0.1577	0.082153	0.105395	8.361	3.11498	9.87748	6.85812
1	32	1	0.14911	67.9292	1	1	1	1	0.097766	0.147628	0.073951	0.093693	7.21346	2.88482	8.28885	5.89864
1	33	1	0.177254	111.714	3.71971	1	1	1	0.089587	0.154615	0.076327	0.103115	8.03919	3.05834	9.33211	7.10189
1	34	1	0.123452	59.3277	1	1	1	1	0.103342	0.149452	0.076773	0.098708	7.92	2.89711	9.01821	6.52277
1	35	1	0.121864	40.389	1	1	1	1	0.094522	0.152446	0.071924	0.102863	7.80095	2.70722	9.75404	6.80225
1	36	1	0.032191	6.13357	1	1	1	1	0.097857	0.149561	0.071791	0.09572	7.6909	2.75067	8.64966	6.25023
1	25	2	0.007812	0.580753	1	1	1	1	0.099515	0.157651	0.08757	0.101517	8.07137	2.6871	9.27442	6.24447
1	26	2	0.023795	6.75246	1	1	1	1	0.097694	0.165553	0.082982	0.103097	8.11531	2.76223	9.28103	6.76821
1	27	2	0.02415	4.32843	1	1	1	1	0.100339	0.158295	0.08292	0.099556	7.98957	2.3562	8.30288	5.91348
1	28	2	0.095907	63.0157	1	1	1	1	0.093487	0.162319	0.082289	0.099777	7.83104	2.5764	8.96572	6.44359
1	29	2	0.166761	212.683	7.81135	1	1	1	0.094699	0.160788	0.085991	0.102983	7.72558	2.75135	8.20318	6.21233
1	30	2	0.783698	1539.39	260.141	7.08837	1	1	0.092476	0.163659	0.084504	0.101034	8.0594	2.57241	8.39289	6.40532
1	31	2	0.634935	1359.45	270.108	3.12731	1	1	0.092863	0.160402	0.081794	0.102312	7.79622	2.64707	8.25038	6.46656
1	32	2	0.397326	891.763	73.066	1.4652	1	1	0.097659	0.163626	0.083329	0.102397	7.93294	2.6261	8.85742	6.50856
1	33	2	0.382958	949.28	69.6777	1	1	1	0.089375	0.157482	0.081622	0.100266	7.63077	2.69438	8.60467	6.33316
1	34	2	0.394203	971.11	54.4961	1	1	1	0.104028	0.167145	0.085124	0.104392	8.50575	3.1765	10.0954	7.38779
1	35	2	0.356317	749.751	46.0135	1	1	1	0.103052	0.163772	0.086259	0.10457	8.54494	3.4276	9.32161	7.27701
1	36	2	0.321815	686.246	20.6516	1	1	1	0.0914	0.162182	0.081523	0.10338	7.96306	2.65373	8.79447	6.34497
1	37	2	0.307555	669.457	14.5127	1	1	1	0.092649	0.164973	0.090239	0.105674	8.52598	3.05213	9.09081	7.00687
1	38	2	0.197655	336.642	4.9712	1	1	1	0.092358	0.167714	0.083181	0.106095	8.64571	2.98266	9.52706	7.13221
1	39	2	0.195454	213.156	3.32063	1	1	1	0.095875	0.159813	0.078469	0.098556	7.61874	2.57563	7.9293	6.07804
1	40	2	0.10796	68.5703	1	1	1	1	0.081696	0.153082	0.069932	0.09838	6.98348	1.96715	7.11342	5.77927
1	41	2	0.024742	9.19288	1	1	1	1	0.100341	0.160218	0.085225	0.103974	8.19121	2.452	9.01469	6.33642
1	42	2	0.008222	3.26444	1	1	1	1	0.085247	0.163458	0.077757	0.098822	7.87887	2.21861	8.13142	5.35075
1	22	3	0.016159	7.3929	1	1	1	1	0.107972	0.157862	0.076304	0.104355	8.224	3.17609	9.78446	4.98496
1	23	3	0.055183	31.17	1	1	1	1	0.082378	0.155237	0.084738	0.105359	8.35206	2.92652	10.4271	6.8831
1	24	3	0.082328	79.4161	1	1	1	1	0.09593	0.156734	0.076042	0.101421	7.90901	2.50096	8.77999	5.88191
1	25	3	0.156418	292.2	1	1	1	1	0.07763	0.159766	0.078562	0.104337	8.08067	2.52481	10.115	6.2989
1	26	3	0.164447	696.991	14.1758	1	1	1	0.118276	0.164563	0.08795	0.103424	8.52946	2.93952	10.0027	6.29202

Fig. 1 CP + WAT Data Format.

4. Getting Started

4.1. User Interface Overview

The main window is divided into several panels, as illustrated in Fig. 2:

I. *Menu*:

Provides access to key functions such as file operations, analysis tools, strategy settings, and help resources.

II. *Data Overview Panel*:

Displays the Lot-Wafer Identification (LWID), providing a summary of the loaded data and key metrics.

III. *Option Panel*:

Enables users to configure specific parameters for their analyses, facilitating customized data exploration.

IV. *Notebook Panel*:

Presents various analysis results and visualizations, allowing users to view and interpret data outputs effectively.

V. *Console Panel*:

Features an integrated Python console that supports advanced operations, enabling users to execute custom scripts and perform complex analyses.

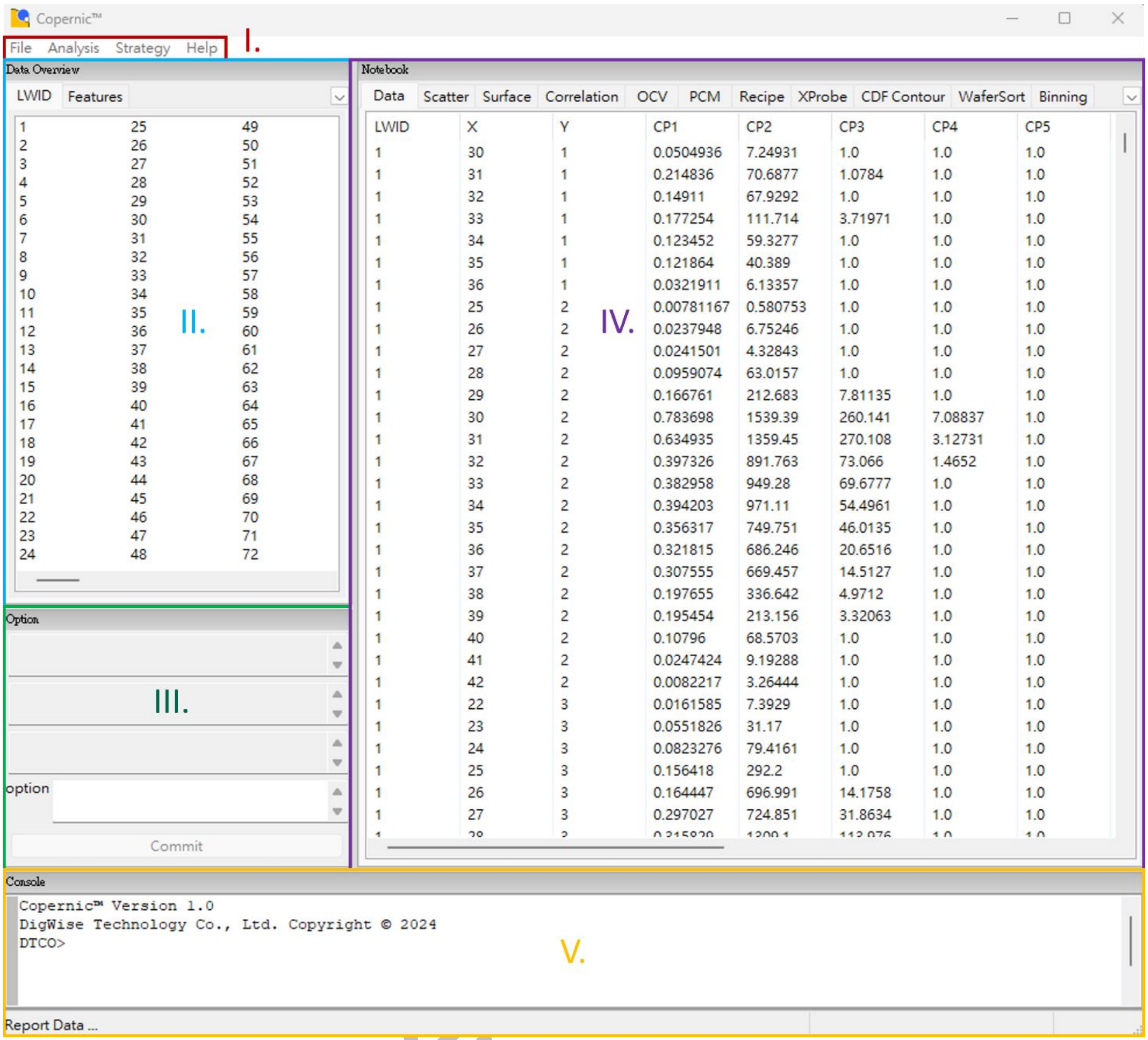

Fig. 2 Copernic™ Main Window.

4.2. Loading Data

To begin utilizing the Copernic™, follow these steps to load your CP (Chip Probe) and WAT (Wafer Test) data:

1. Navigate to Load Data Option:
   - Click on **"File"** in the top menu, then select "**Load Data"** from the dropdown menu.
2. Select Your Files:
   - In the file dialog that appears, choose the appropriate CP and WAT data files that you wish to import. You can select multiple files if necessary.
3. Process the Data:
   - Once you have selected the files, the Copernic™ will process the data automatically. After processing, **Data Overview Panel** will update to display the imported information, as illustrated in Fig. 3.

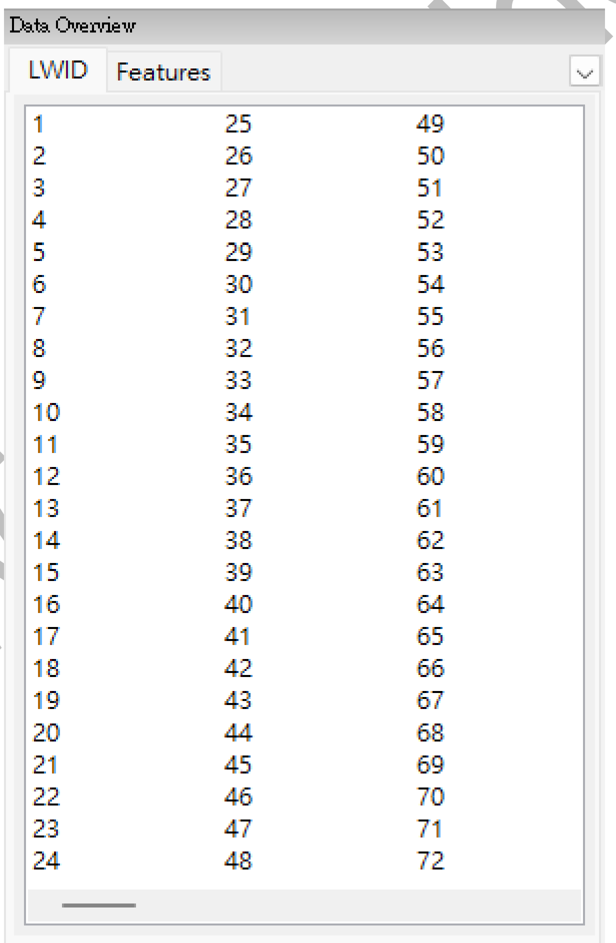

Fig. 3 Data Overview Panel display imported LWID information.

4.3. LWID Selection

In the Copernic™, users can select LWID numbers in the **LWID Panel** through three different methods, allowing for flexible selection based on user preferences:

1. **LWID Single Selection**:
    - Left-click on the desired LWID number to highlight it. Once highlighted, right-click to display a context menu. From the menu, select **"Choose LWID"** to finalize your selection.

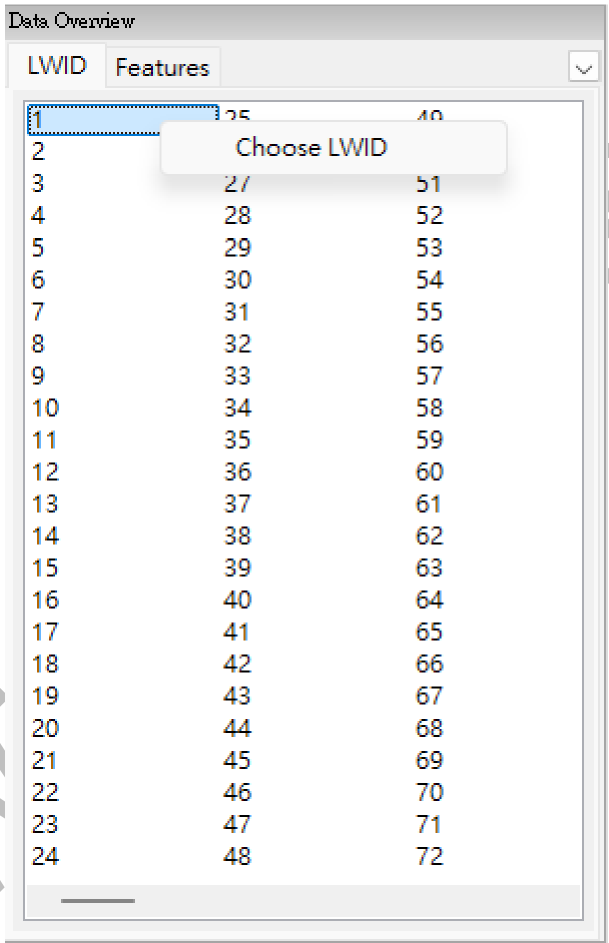

Fig. 4 LWID Single Selection.

2. **LWID Multiple Selection (Ctrl)**:

- Hold down the **Ctrl** key and left-click on multiple LWID numbers to select them. Each selected LWID will be highlighted. After selecting the desired LWIDs, right-click and choose **"Choose LWID"** from the context menu.

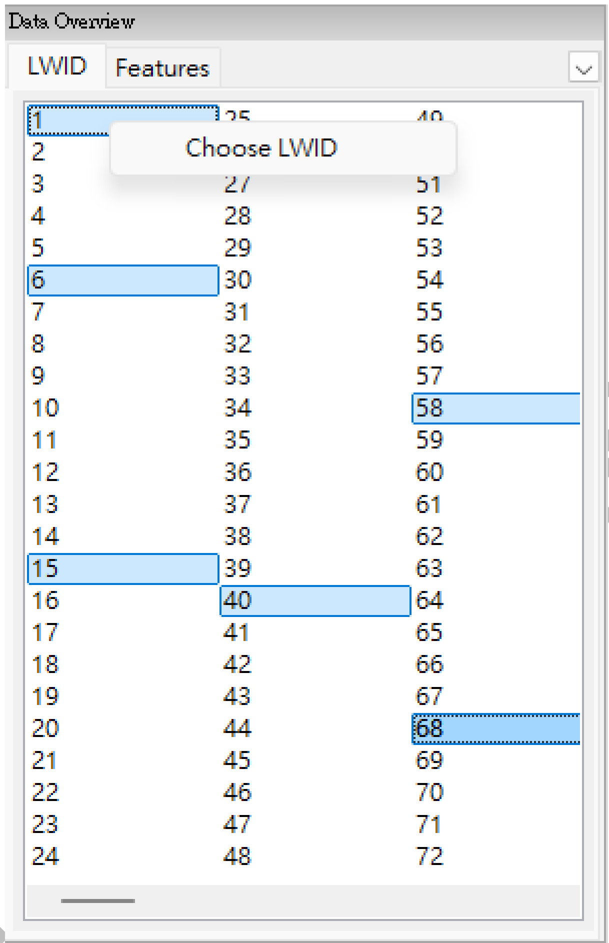

Fig. 5 LWID Multiple Selection.

3. **LWID Range Selection (Shift)**:

   - Hold down the **Shift** key, left-click on the first LWID (the head of the range), and then left-click on another LWID (the tail of the range). This will highlight the entire range of LWID numbers. Right-click anywhere within the selected range and choose **"Choose LWID"** from the context menu to complete the selection.

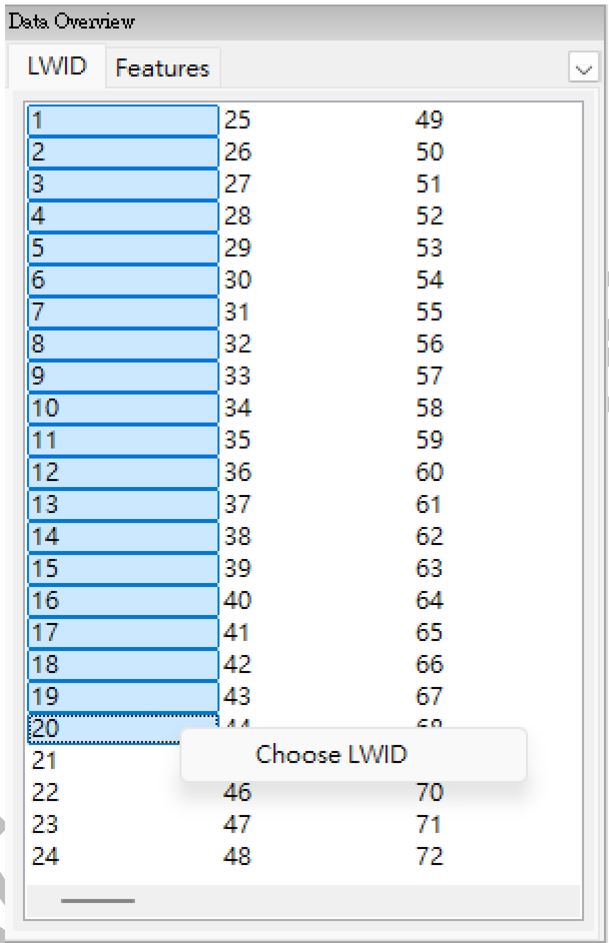

Fig. 6 LWID Range Selection.

These flexible selection methods allow users to efficiently choose one or more LWID numbers, adapting to different use cases such as individual or batch selections.

After completing any of the above LWID selection methods, the **Data Overview Panel** will automatically update to display the corresponding features information, allowing for further data analysis, as illustrated in Fig. 6.

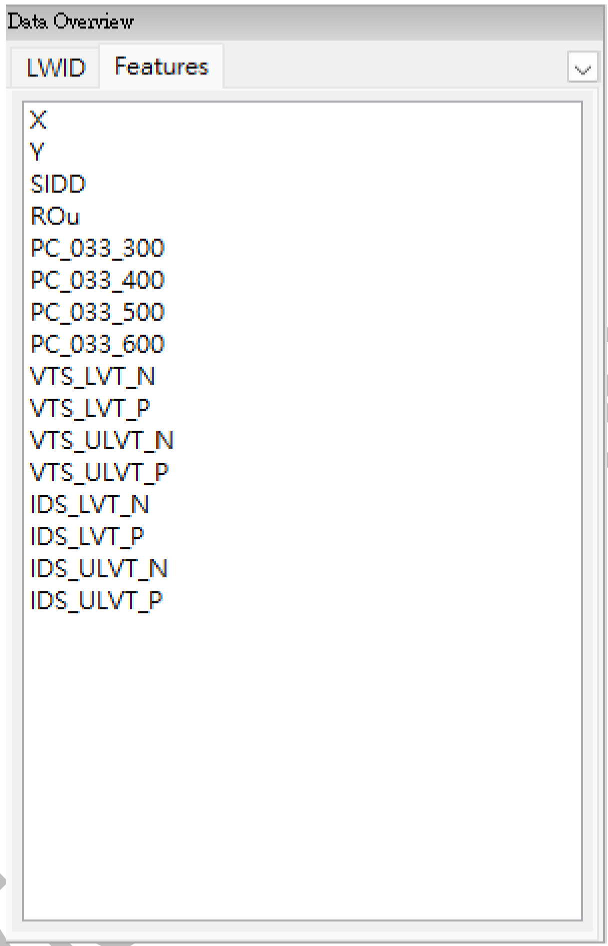

Fig. 7 Display features information after LWID Selection.

Additionally, the **"Data"** tab under the Notebook Panel will simultaneously display the DataFrame corresponding to the selected LWID numbers, reflecting the updates based on the recent LWID selection, as illustrated in Fig. 8.

Notebook

Data | Scatter | Surface | Correlation | OCV | PCM | Recipe | XProbe | CDF Contour | WaferSort | Binning

LWID	X	Y	SIDD	ROu	PC_033_300	PC_033_400	PC_033_500
1	30	1	0.0504936	7.24931	1.0	1.0	1.0
1	31	1	0.214836	70.6877	1.0784	1.0	1.0
1	32	1	0.14911	67.9292	1.0	1.0	1.0
1	33	1	0.177254	111.714	3.71971	1.0	1.0
1	34	1	0.123452	59.3277	1.0	1.0	1.0
1	35	1	0.121864	40.389	1.0	1.0	1.0
1	36	1	0.0321911	6.13357	1.0	1.0	1.0
1	25	2	0.00781167	0.580753	1.0	1.0	1.0
1	26	2	0.0237948	6.75246	1.0	1.0	1.0
1	27	2	0.0241501	4.32843	1.0	1.0	1.0
1	28	2	0.0959074	63.0157	1.0	1.0	1.0
1	29	2	0.166761	212.683	7.81135	1.0	1.0
1	30	2	0.783698	1539.39	260.141	7.08837	1.0
1	31	2	0.634935	1359.45	270.108	3.12731	1.0
1	32	2	0.397326	891.763	73.066	1.4652	1.0
1	33	2	0.382958	949.28	69.6777	1.0	1.0
1	34	2	0.394203	971.11	54.4961	1.0	1.0
1	35	2	0.356317	749.751	46.0135	1.0	1.0
1	36	2	0.321815	686.246	20.6516	1.0	1.0
1	37	2	0.307555	669.457	14.5127	1.0	1.0
1	38	2	0.197655	336.642	4.9712	1.0	1.0
1	39	2	0.195454	213.156	3.32063	1.0	1.0
1	40	2	0.10796	68.5703	1.0	1.0	1.0
1	41	2	0.0247424	9.19288	1.0	1.0	1.0
1	42	2	0.0082217	3.26444	1.0	1.0	1.0
1	22	3	0.0161585	7.3929	1.0	1.0	1.0
1	23	3	0.0551826	31.17	1.0	1.0	1.0
1	24	3	0.0823276	79.4161	1.0	1.0	1.0
1	25	3	0.156418	292.2	1.0	1.0	1.0
1	26	3	0.164447	696.991	14.1758	1.0	1.0
1	27	3	0.297027	724.851	31.8634	1.0	1.0

Fig. 8 Display data information after LWID Selection.

4.4. Features Selection

In the Copernic™, users can select features in the **Features Panel** using three different methods, such as Scatter Analysis, providing flexible selection based on user preferences:

1. **Feature Single Selection**:
   - Select the desired feature with a left-click to highlight it. After highlighting, click and hold the left mouse button, then drag the selection to the X, Y, and Z rows in the **Option Panel** below, and release to finalize your selection.

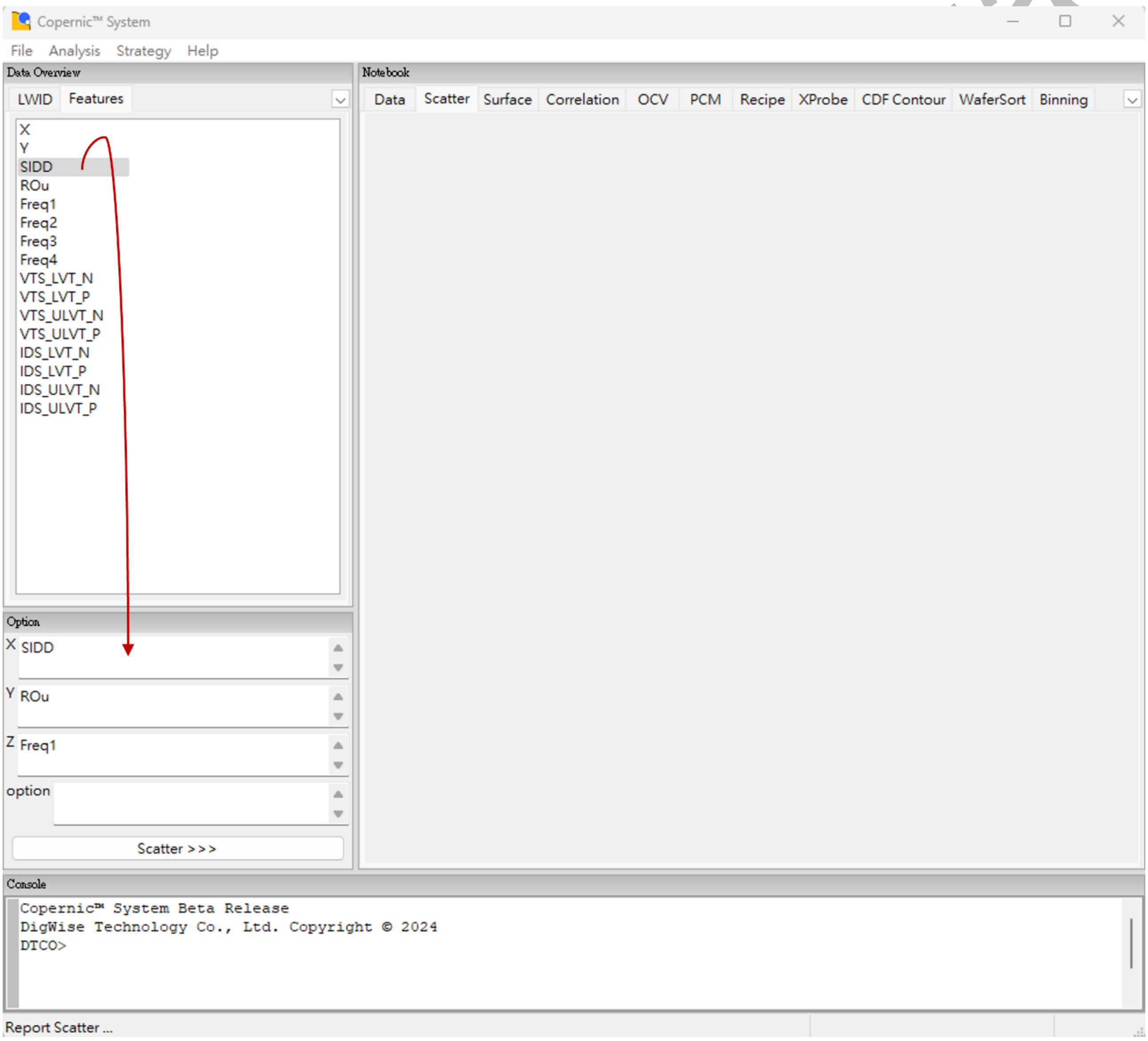

Fig. 9 Feature Single Selection.

2. **Features Multiple Selection (Ctrl)**:

   - Hold down the **Ctrl** key and left-click on multiple features to highlight them. After highlighting, click and hold the left mouse button, then drag the selection to the X, Y, and Z rows in the **Option Panel** below, and release to finalize your selection.

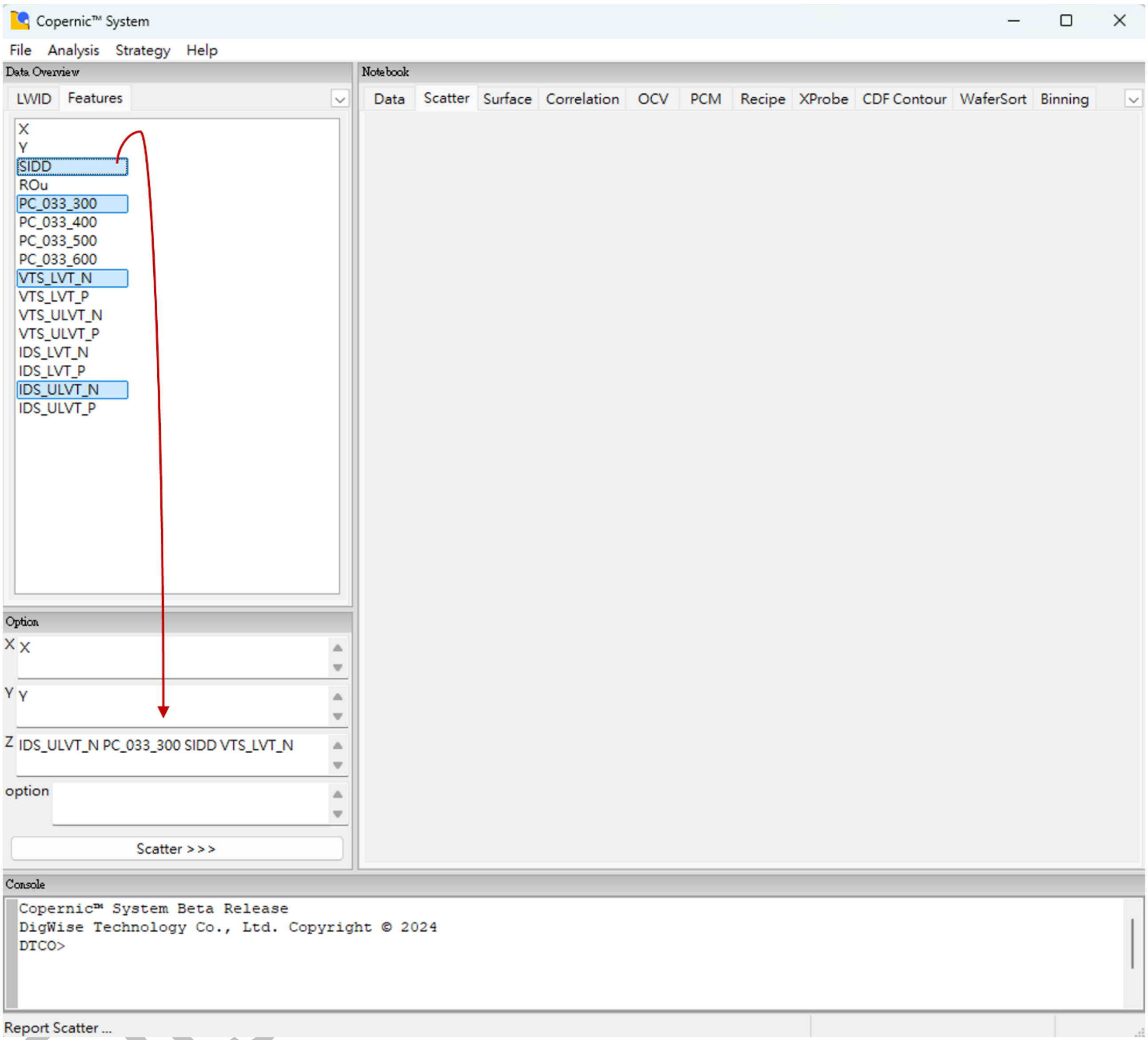

Fig. 10 Features Multiple Selection.

3. **Features Range Selection (Shift)**:

   - Hold down the **Shift** key and left-click on the first feature (the head of the range), and then left-click on another feature (the tail of the range). This will highlight the entire range of features. After highlighting, click and hold the left mouse button, then drag the selection to the X, Y, and Z rows in the **Option Panel** below, and release to finalize your selection.

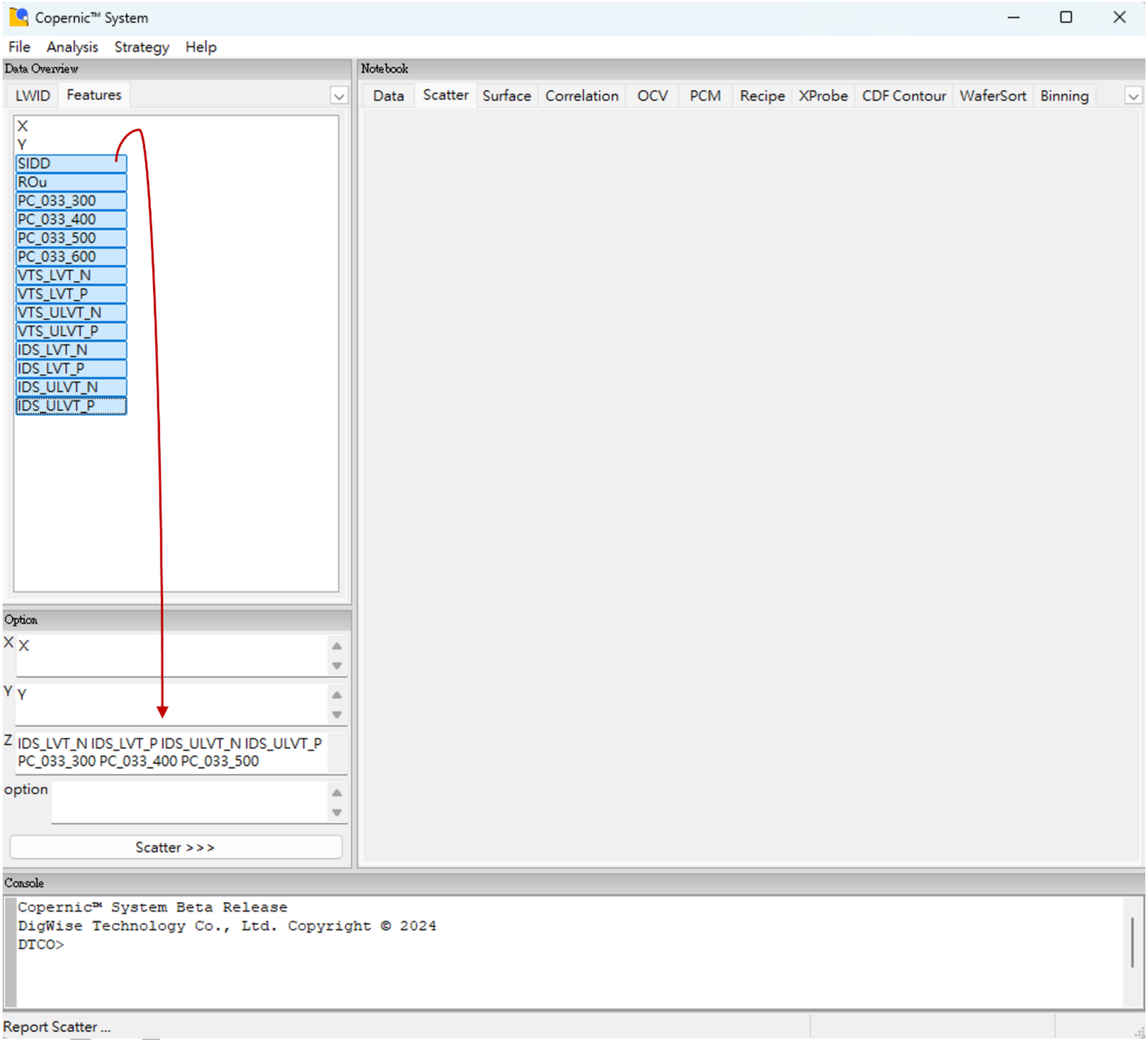

Fig. 11 Features Range Selection.

4.5. Constraint Parameters

In the Copernic™, users can modify constraint parameters in the **Option Panel**, like filtering, such as Scatter Analysis, refine your analysis according to specific criteria.

1. Filtering in the **Option Panel**:
   - The **Option Panel** allows for advanced data filtering using the keyword sigma (ex: sigma = 2). To apply a sigma-based filter, input the desired sigma value.

(1) No Filtered Data:

Do not input sigma, red circle are outliers.

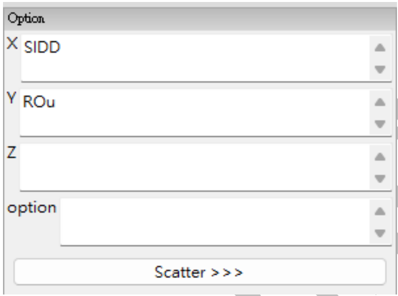

Fig. 12 Do not input sigma.

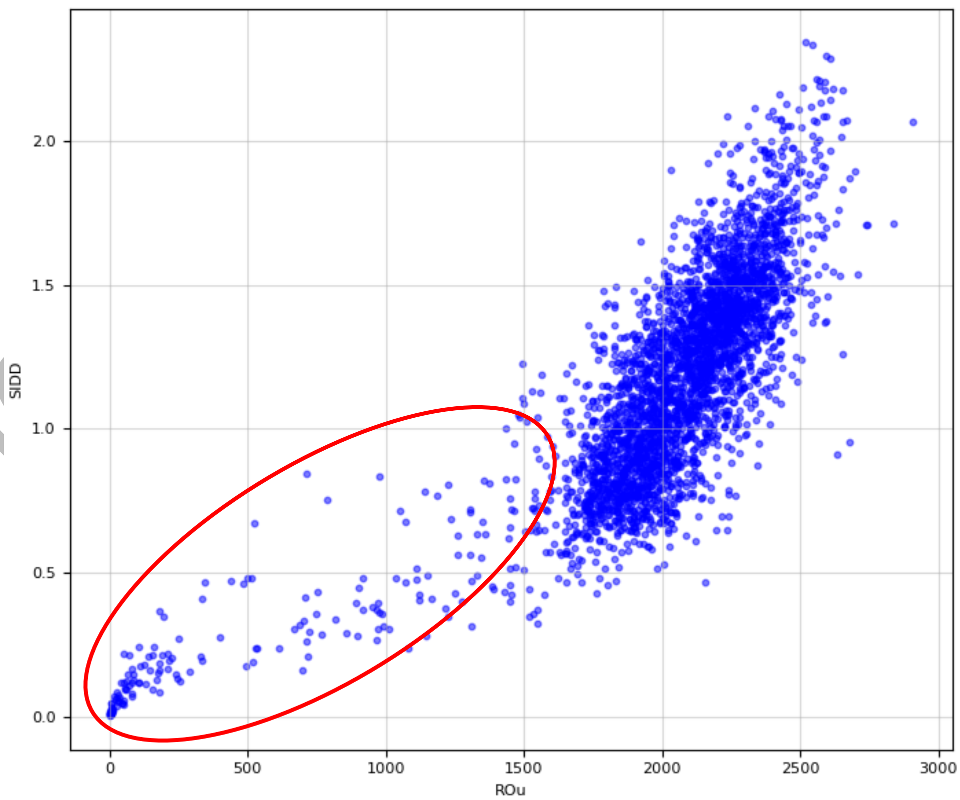

Fig. 13 ROu vs SIDD scatter (include outliers).

(2) Filtered Data:

Input the required sigma value to apply the filter, which will restrict the dataset based on the specified sigma threshold.

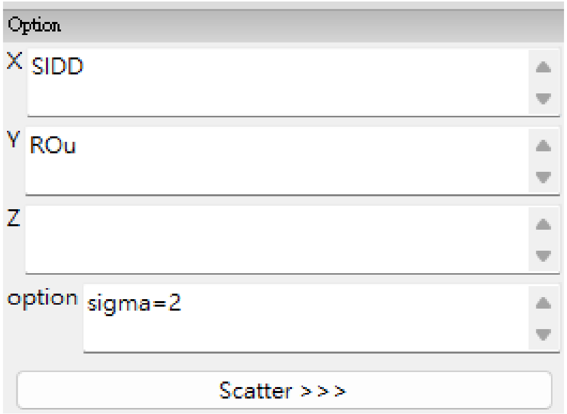

Fig. 14 Input sigma.

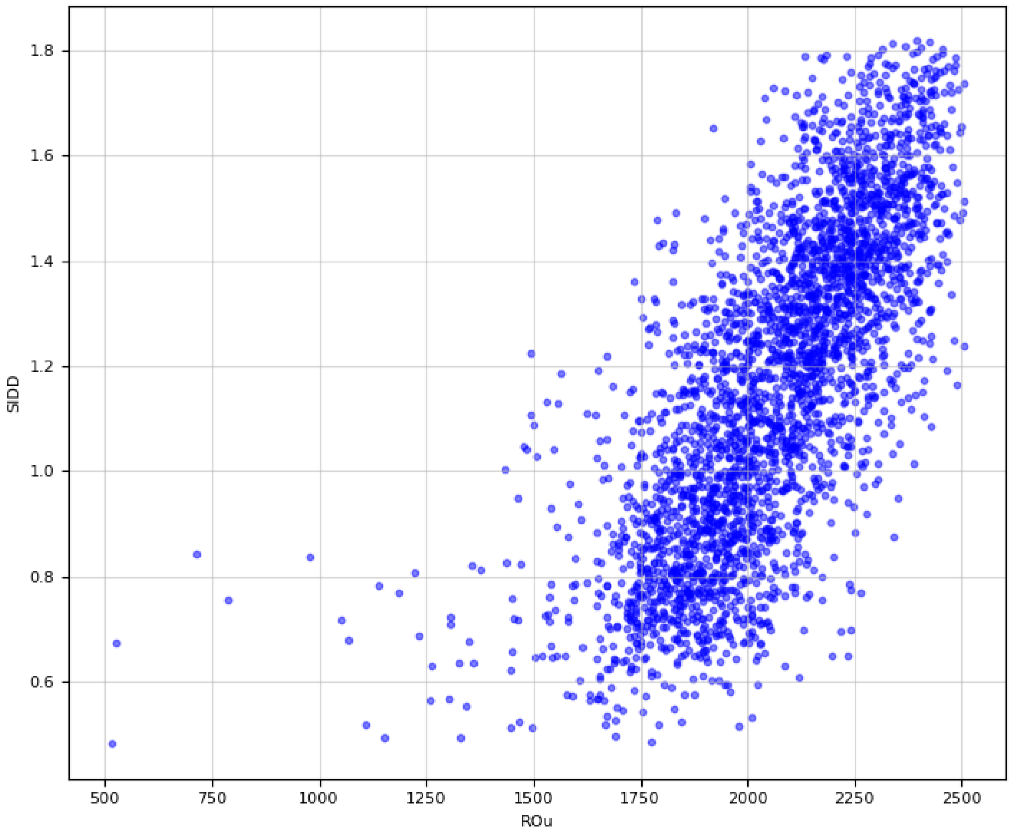

Fig. 15 ROu vs SIDD scatter (drop outliers).

(3) Sigma Definition:

Sigma (σ) represents the standard deviation in a Gaussian (normal) distribution, quantifying the amount of variation or dispersion from the mean. It indicates how spread out the data points are: a smaller σ means the data points are closer to the mean, while a larger σ indicates they are more spread out. In a Gaussian distribution, approximately 68% of the data falls within ±1σ, 95% within ±2σ, and 99.7% within ±3σ of the mean.

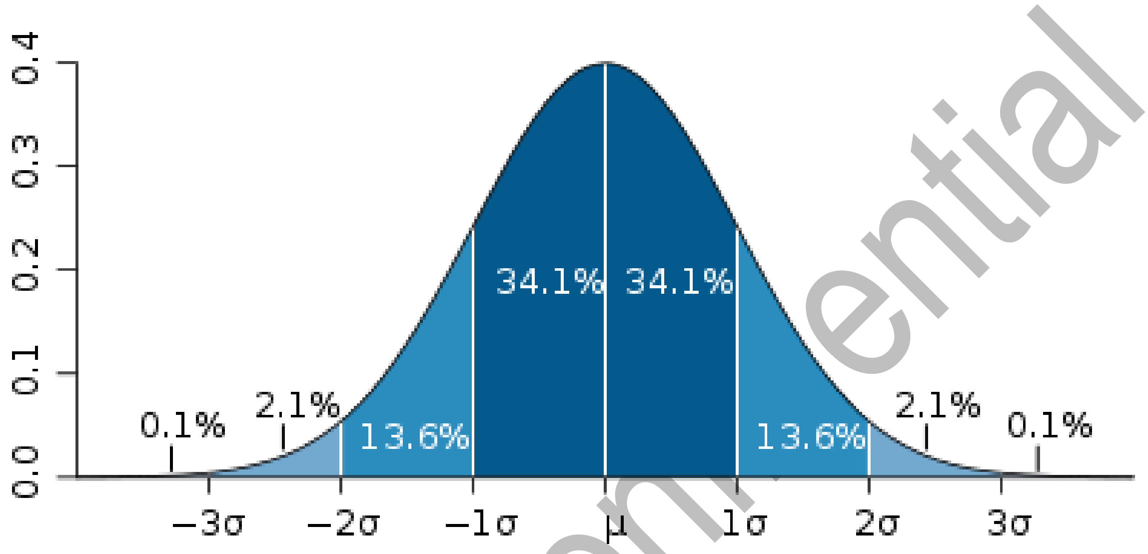

Fig. 16 Gaussian normal distribution.

5. Analysis Procedures

5.1. Batch View (2D/3D)

1. Generate Results:
   - After loading the data, left-click on **"Analysis"**, then select **"Batch View"** from the dropdown menu. Choose either **"2D"** or **"3D"**, and left-click to generate the analysis results for review.

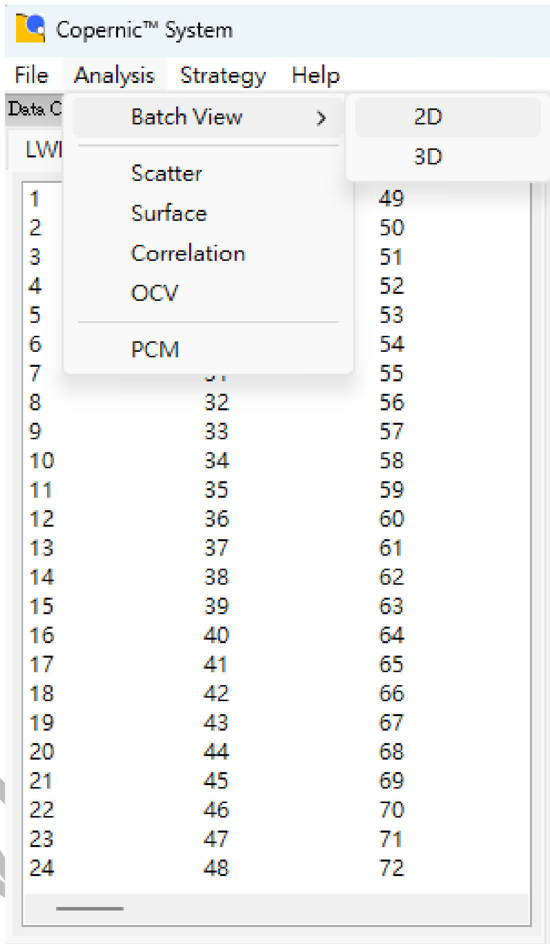

Fig. 17 Batch View analysis operation steps.

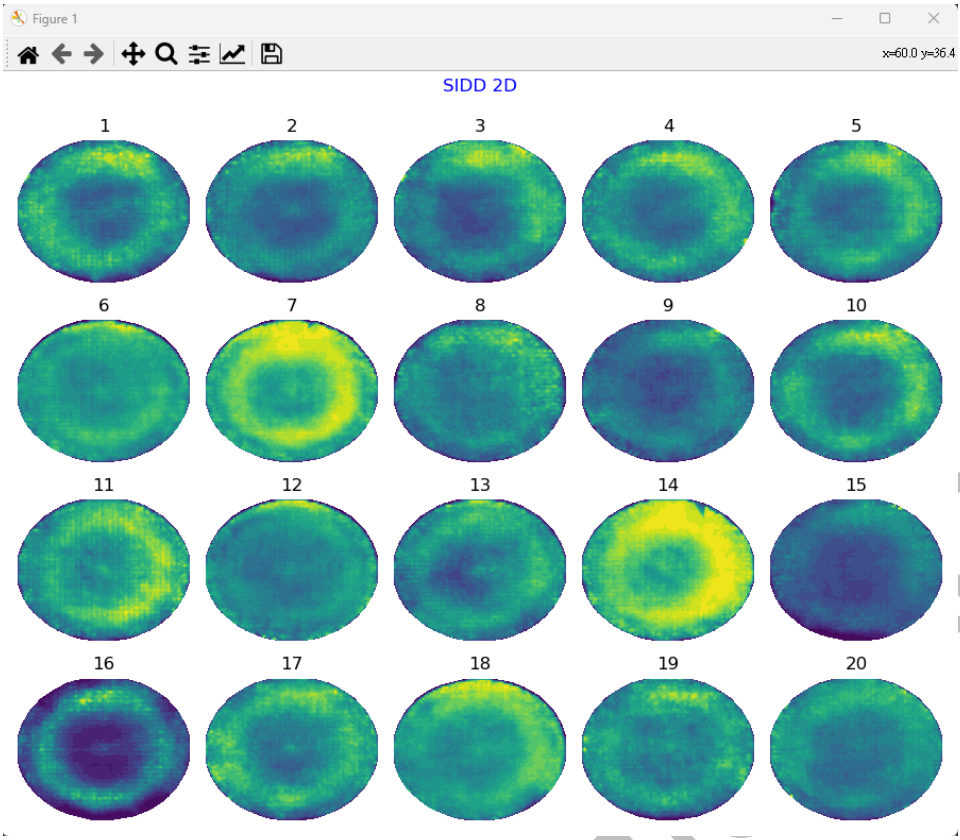

Fig. 18 SIDD 2D Batch View analysis (show pre 20 wafers).

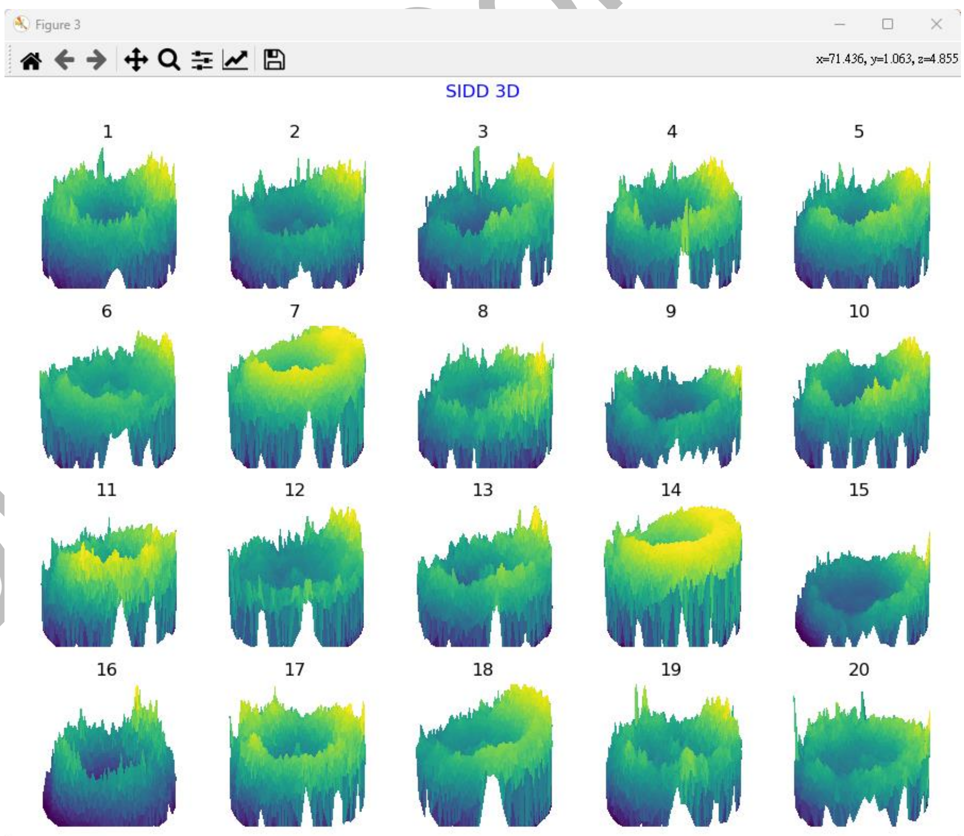

Fig. 19 SIDD 3D Batch View analysis (show pre 20 wafers).

5.2. Scatter Analysis

1. Access the Scatter Tab:
   - Navigate to the **"Scatter"** tab located within the **Notebook Panel** to initiate the analysis.
2. Select LWID and Features:
   - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features you wish to analyze.
3. Adjust Parameters:
   - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
   - Click the **"Scatter >>>"** button to execute the scatter analysis, which will produce the analysis results for review.

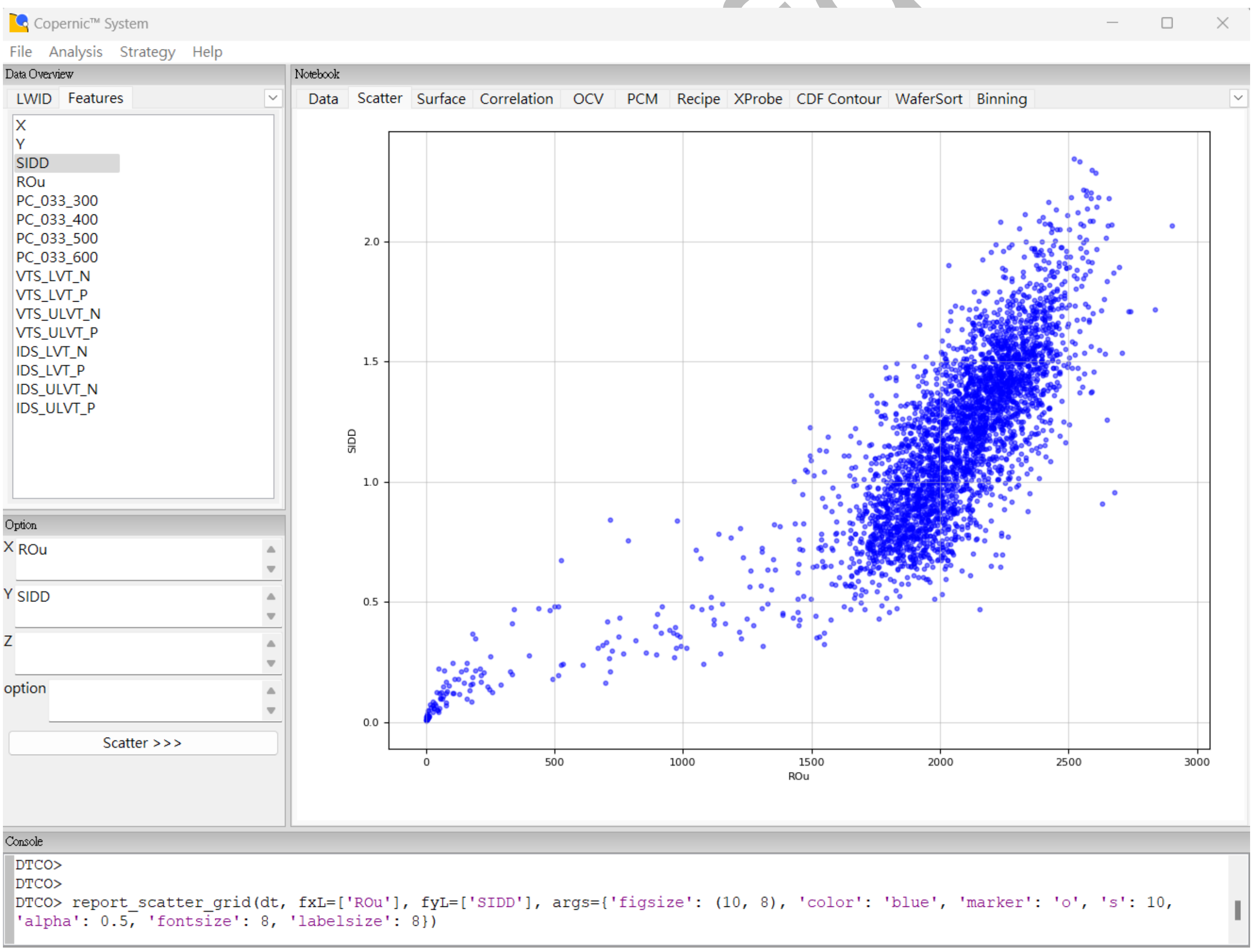

Fig. 20 ROu vs SIDD Scatter Analysis (for example).

5.2. Surface Analysis

1. Select LWID and Features:
   - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features you wish to analyze.
2. Access the Surface Tab:
   - Navigate to the **"Surface"** tab located within the **Notebook Panel** to initiate the analysis.
3. Adjust Parameters:
   - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
   - Click the **"Surface >>>"** button to execute the surface analysis, which will produce the analysis results for review.

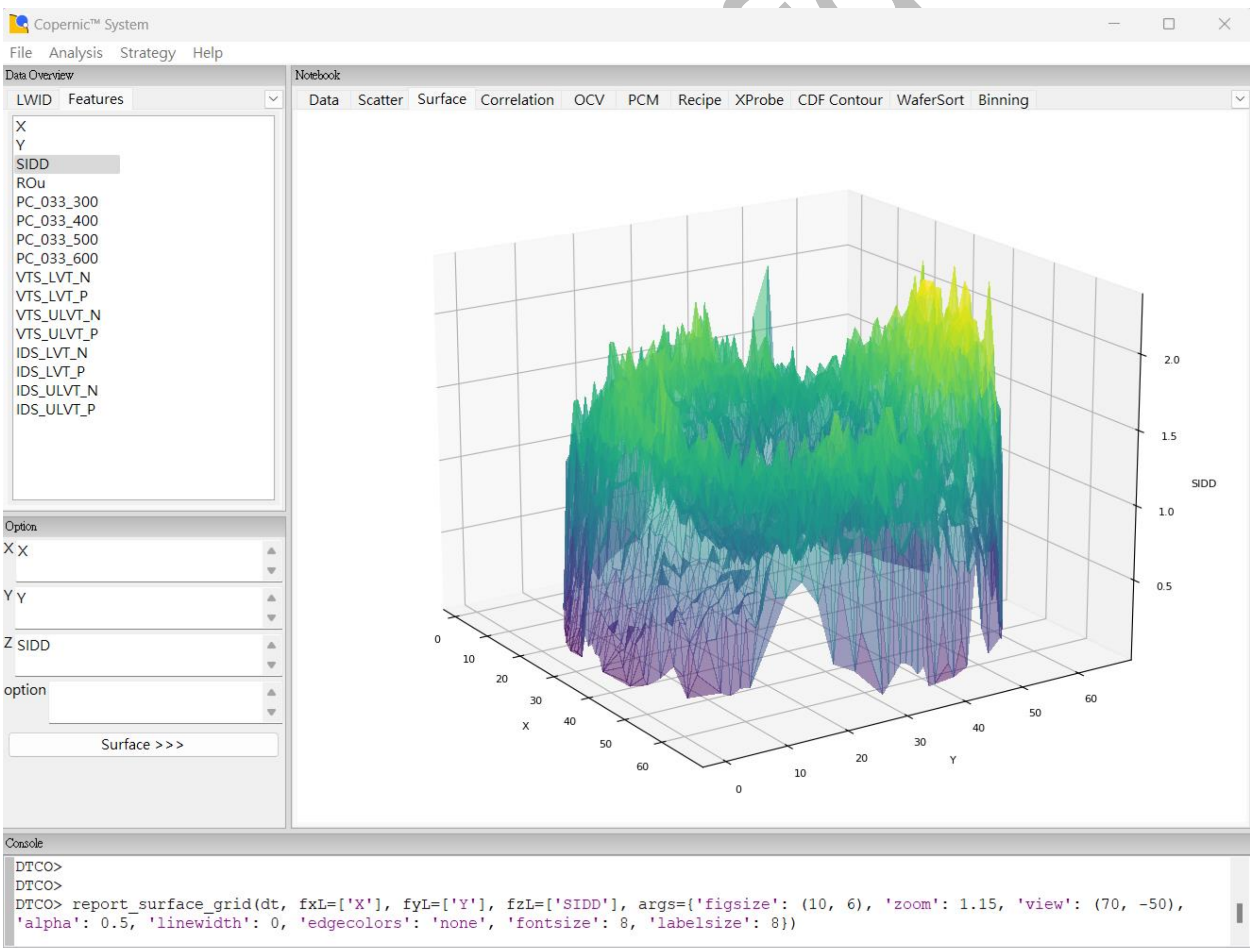

Fig. 21 SIDD Surface Analysis (for example).

5.3. Correlation Analysis

1. Select LWID and Features:
   - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features (at least 2) you wish to analyze.
2. Access the Correlation Tab:
   - Navigate to the **"Correlation"** tab located within the **Notebook Panel** to initiate the analysis.
3. Adjust Parameters:
   - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
   - Click the **"Correlation >>>"** button to execute the correlation analysis, which will produce the analysis results for review.

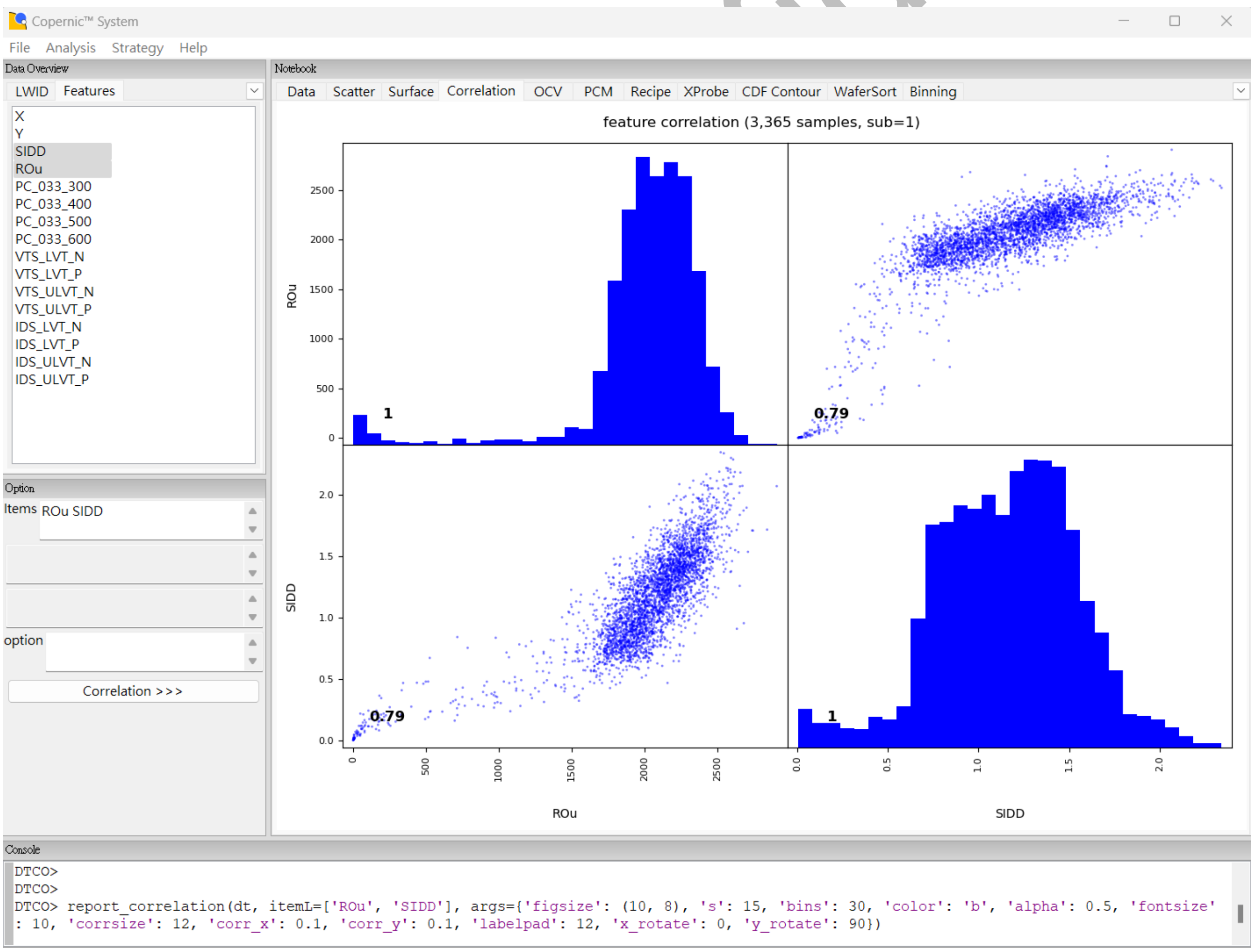

Fig. 22 SIDD vs ROu Correlation Analysis (for example).

5.4. OCV Analysis

1. Select LWID and Features:
   - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features (default: ROu) you wish to analyze.
2. Access the OCV Tab:
   - Navigate to the **"OCV"** tab located within the **Notebook Panel** to initiate the analysis.
3. Adjust Parameters:
   - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
   - Click the **"OCV >>>"** button to execute the OCV analysis, which will produce the analysis results for review.

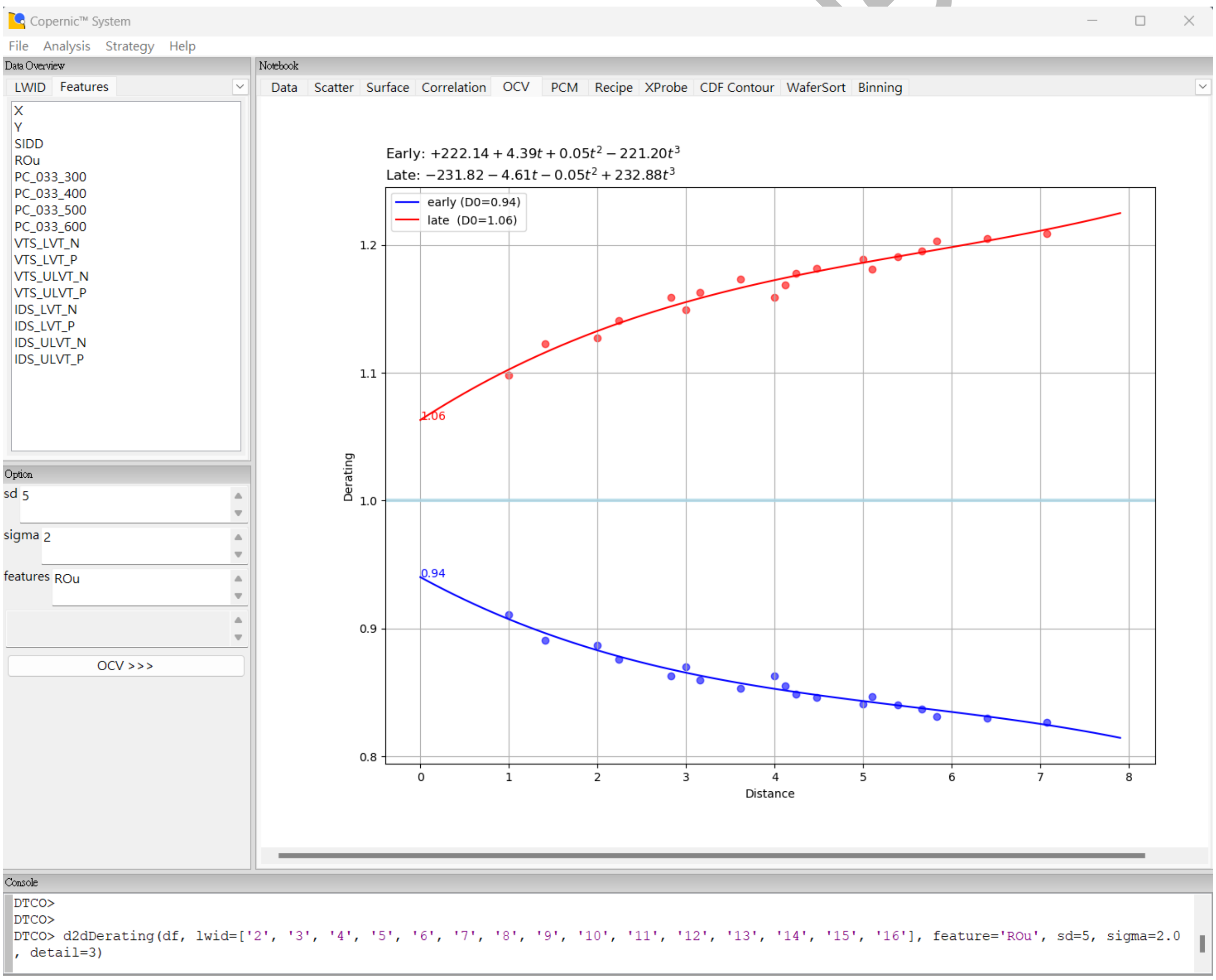

Fig. 23 ROu OCV Analysis (for example).

5.5. PCM Analysis

1. Select LWID and Features:
    - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features you wish to analyze.
2. Access the PCM Tab:
    - Navigate to the **"PCM"** tab located within the **Notebook Panel** to initiate the analysis.
3. Adjust Parameters:
    - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
    - Click the **"PCM >>>"** button to execute the PCM analysis, which will produce the analysis results for review.

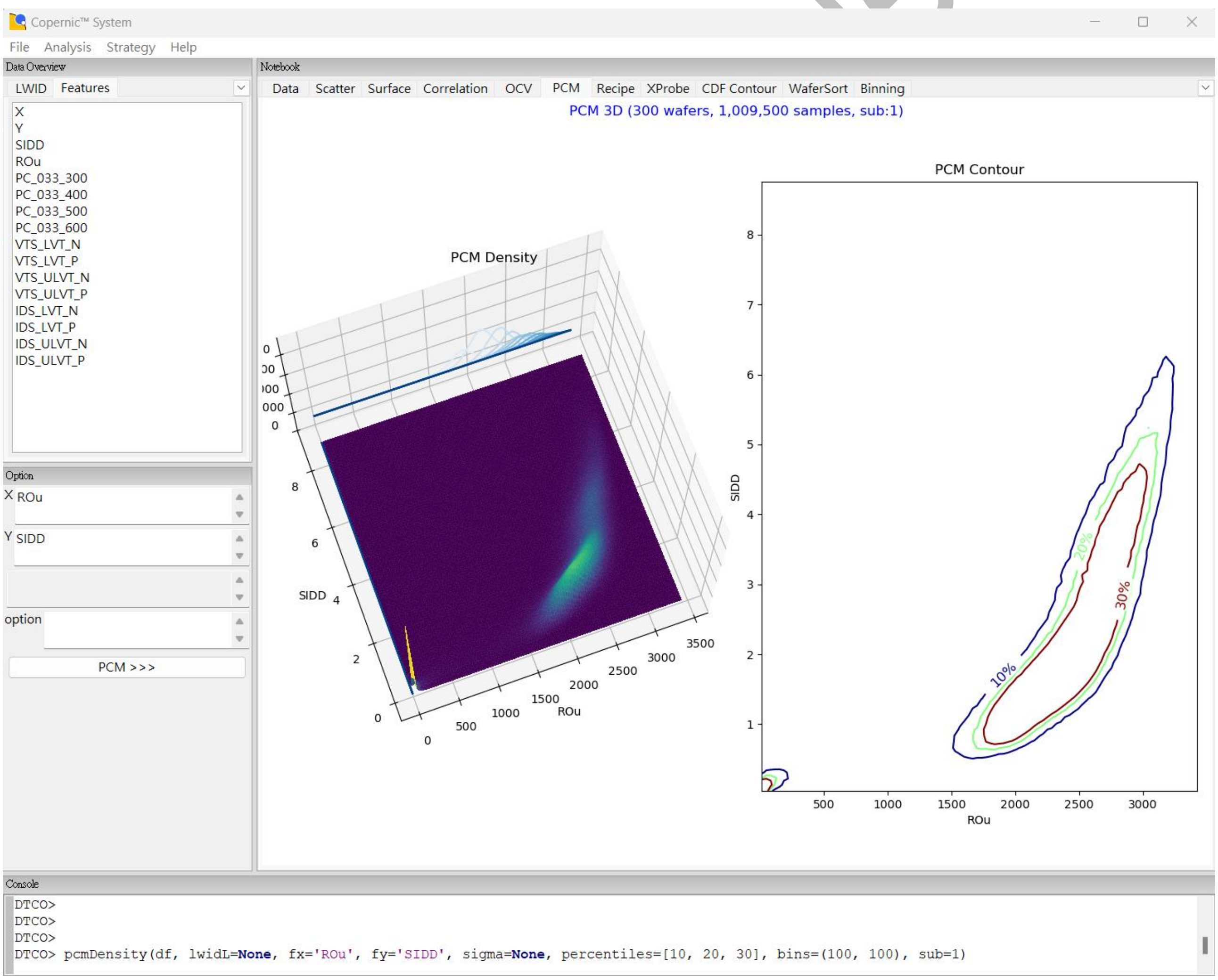

Fig. 24 ROu vs SIDD PCM Analysis (for example).

5.6. Recipe Analysis

1. Select LWID and Features:
   - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features you wish to analyze.
2. Access the Recipe Tab:
   - Navigate to the **"Recipe"** tab located within the **Notebook Panel** to initiate the analysis.
3. Adjust Parameters:
   - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
   - Click the **"Recipe >>>"** button to execute the recipe analysis, which will produce the analysis results for review.

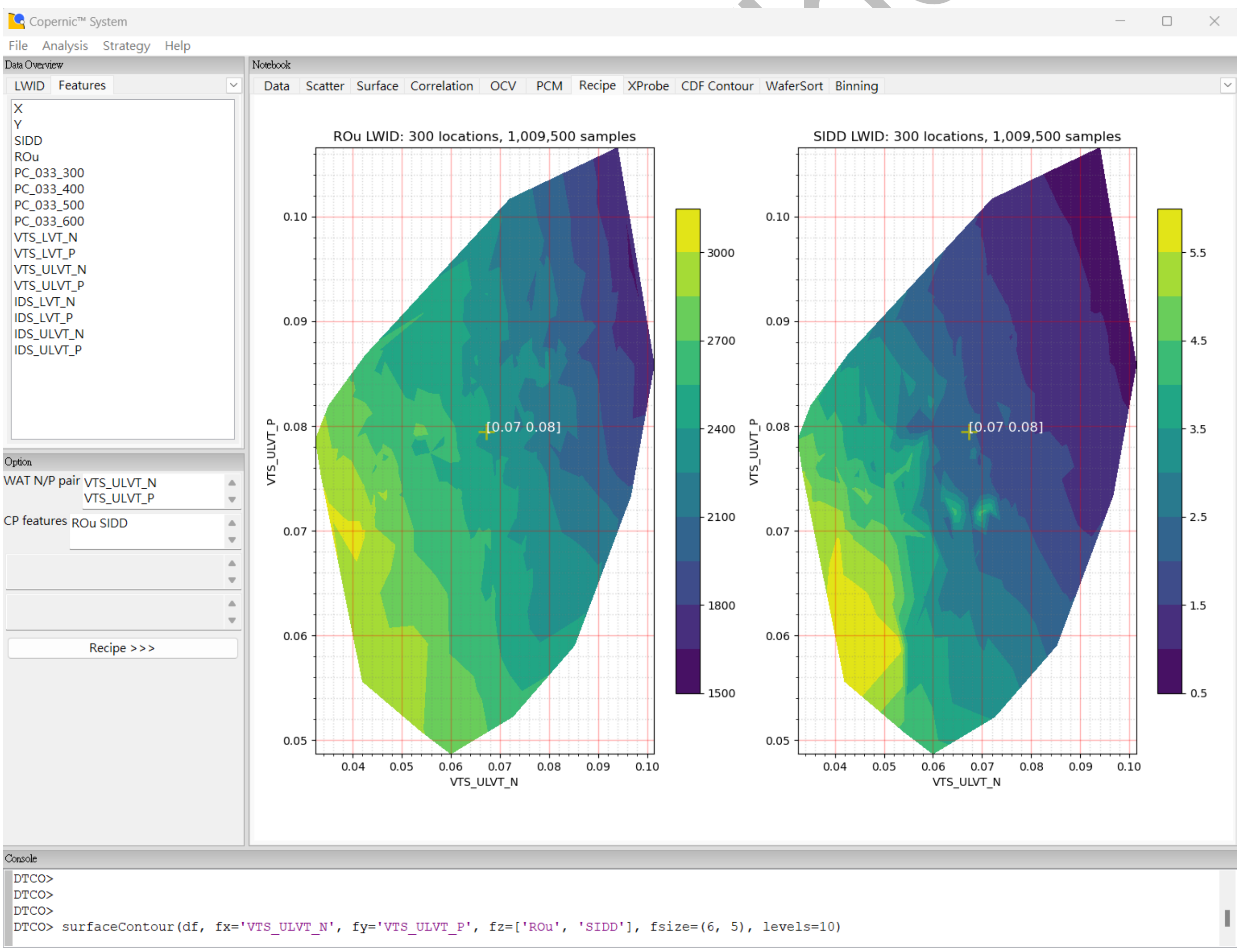

Fig. 25 CP (ROu, SIDD) vs WAT (VTS_ULVT_N/P) Recipe Analysis (for example).

5.7. Xprobe Analysis

1. Select LWID and Features:
   - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features you wish to analyze.
2. Access the XProbe Tab:
   - Navigate to the **"XProbe"** tab located within the **Notebook Panel** to initiate the analysis.
3. Adjust Parameters:
   - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
   - Click the **"XProbe >>>"** button to execute the Xprobe analysis, which will produce the analysis results for review.

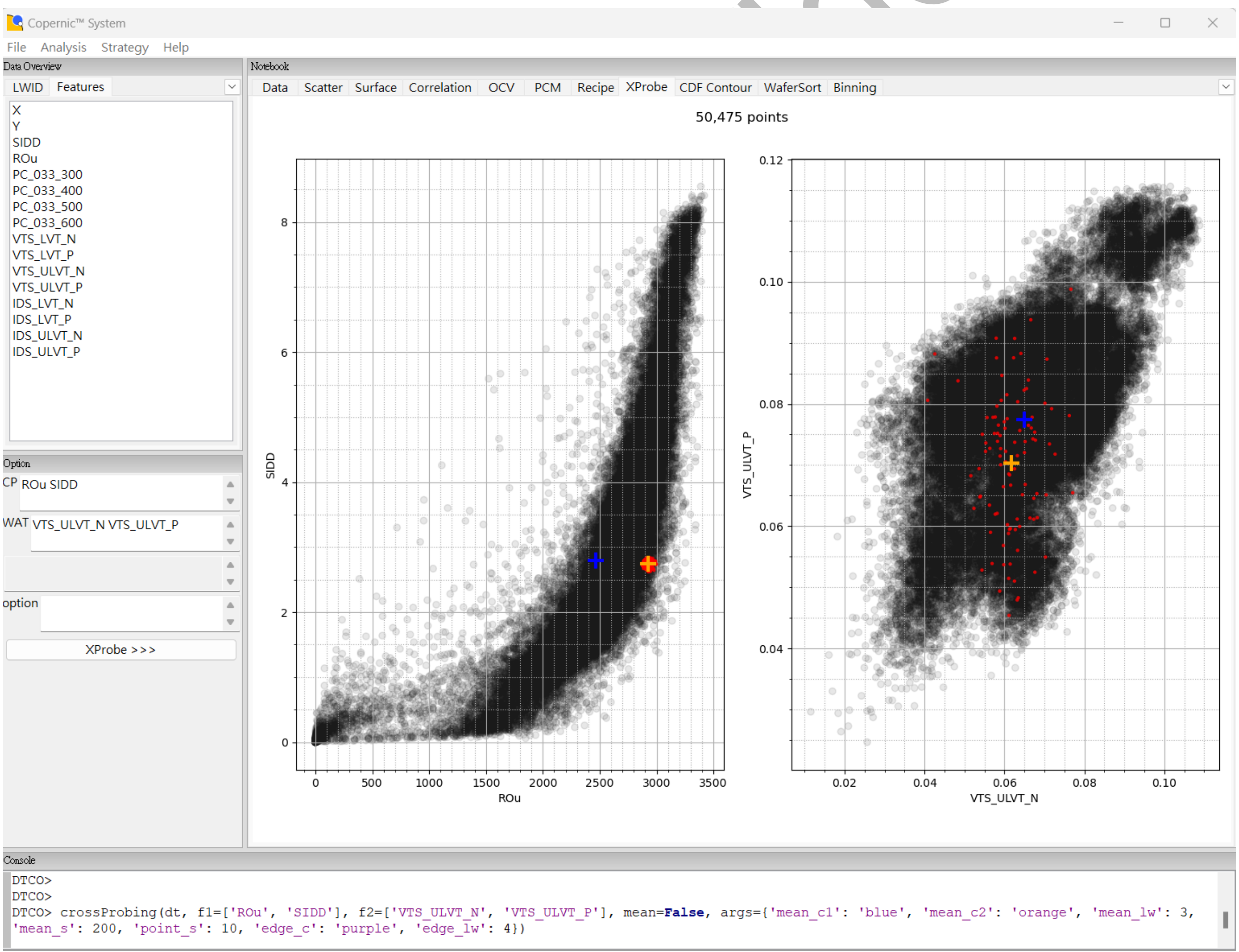

Fig. 26 CP (ROu, SIDD) vs WAT (VTS_ULVT_N/P) Xprobe Analysis (for example)

5.8. CDF Contour Analysis

1. Select LWID and Features:
   - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features you wish to analyze.
2. Access the CDF Contour Tab:
   - Navigate to the **"CDF Contour"** tab located within the **Notebook Panel** to initiate the analysis.
3. Adjust Parameters:
   - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
   - Click the **"CDF Contour >>>"** button to execute the CDF Contour analysis, which will produce the analysis results for review.

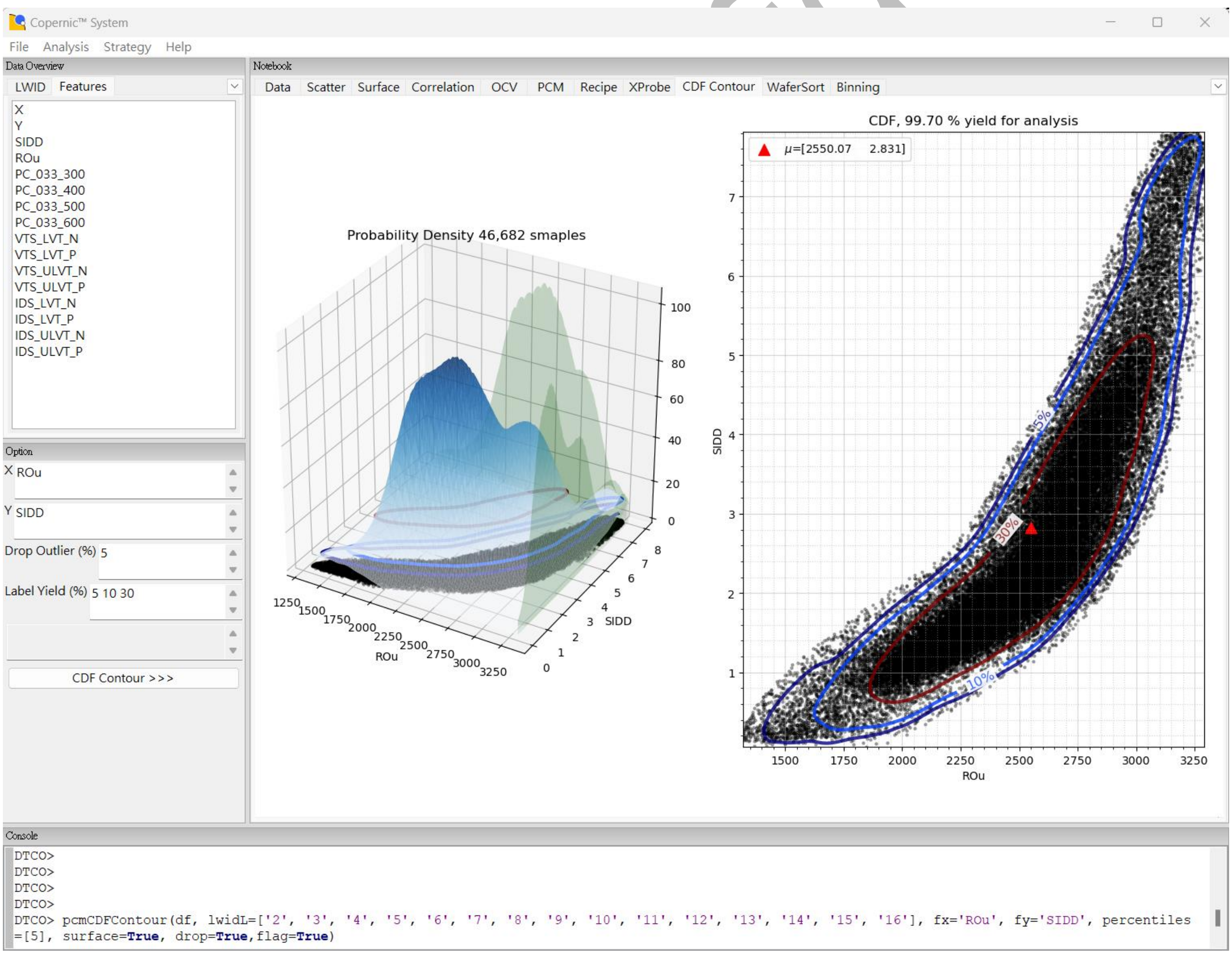

Fig. 27 ROu vs SIDD CDF Contour Analysis (for example).

5.9. WaferSort Analysis

1. Select LWID and Features:
   - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features you wish to analyze.
2. Access the WaferSort Tab:
   - Navigate to the **"WaferSort"** tab located within the **Notebook Panel** to initiate the analysis.
3. Adjust Parameters:
   - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
   - Click the **"WaferSort >>>"** button to execute the WaferSort analysis, which will produce the analysis results for review.

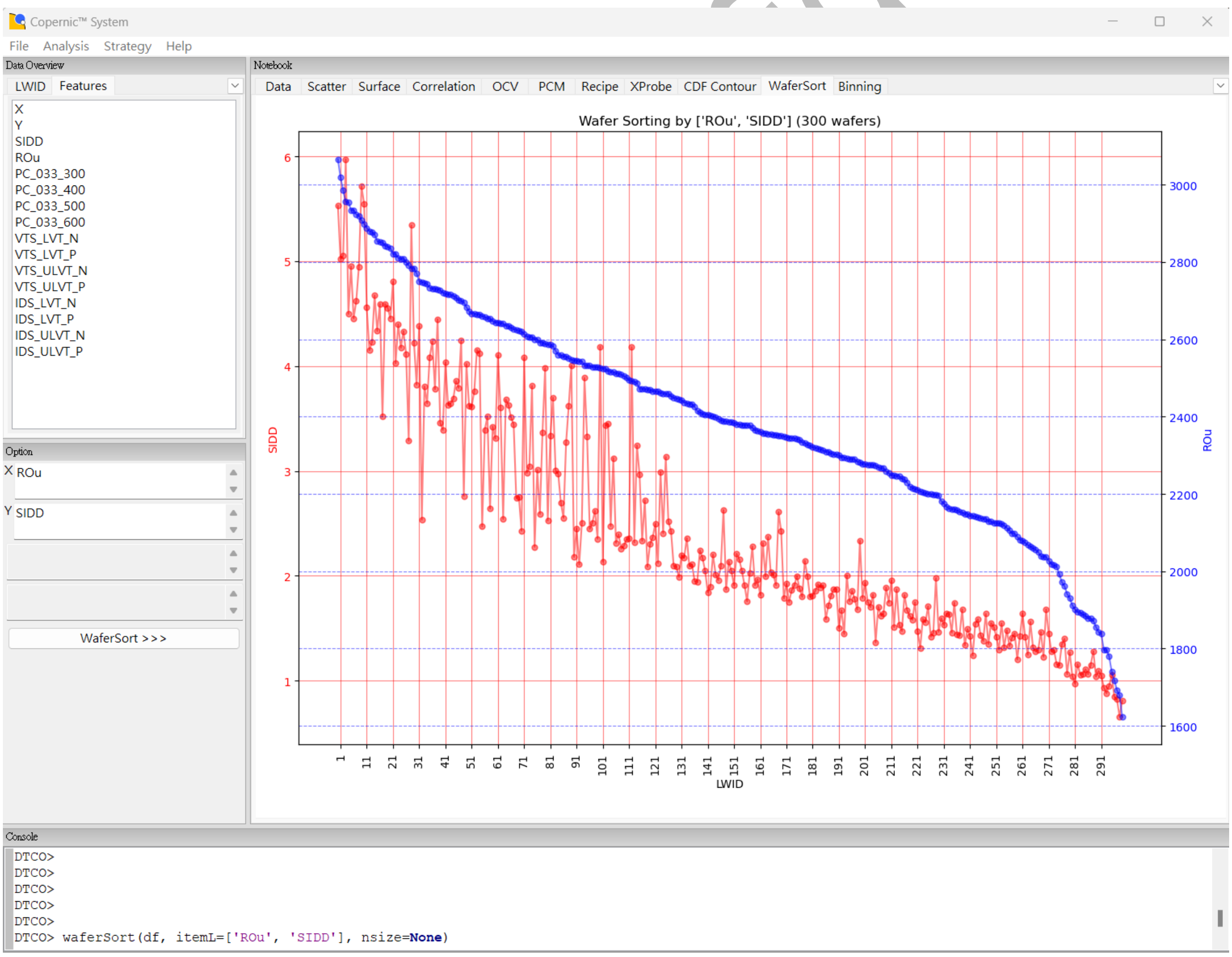

Fig. 28 ROu vs SIDD WaferSort Analysis (for example).

5.10. Binning Analysis

1. Select LWID and Features:
   - In the **Data Overview Panel**, choose the desired Lot-Wafer Identification (LWID) and relevant features you wish to analyze.
2. Access the Binning Tab:
   - Navigate to the **"Binning"** tab located within the **Notebook Panel** to initiate the analysis.
3. Adjust Parameters:
   - If necessary, modify the constraint parameters in the **Option Panel** to refine your analysis according to specific criteria.
4. Generate Results:
   - Click the **"Binning >>>"** button to execute the Binning analysis, which will produce the analysis results for review.

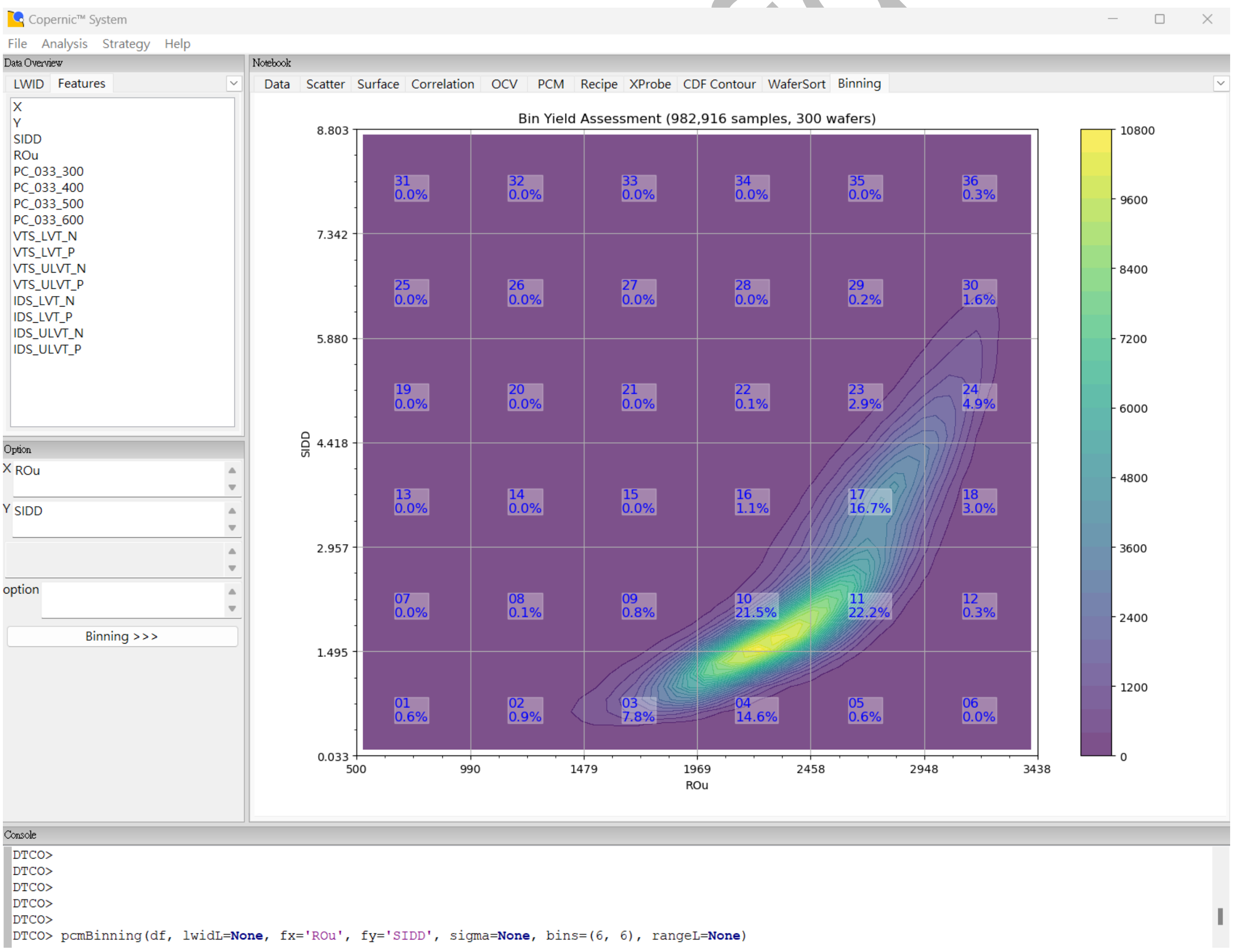

Fig. 29 ROu vs SIDD Binning Analysis (for example).

6. Advanced Application

The Console Panel provides a Python interface for advanced operations:

6.1. Execute custom scripts for specialized analyses

Example Code:

```
report_scatter_grid(dt,fxL=['ROu'],fyL=['SIDD'],args={'figsize':(10,8),'c
olor':'green','marker':'*','s':10,'alpha':0.5,'fontsize':8,'labelsize':8})
```

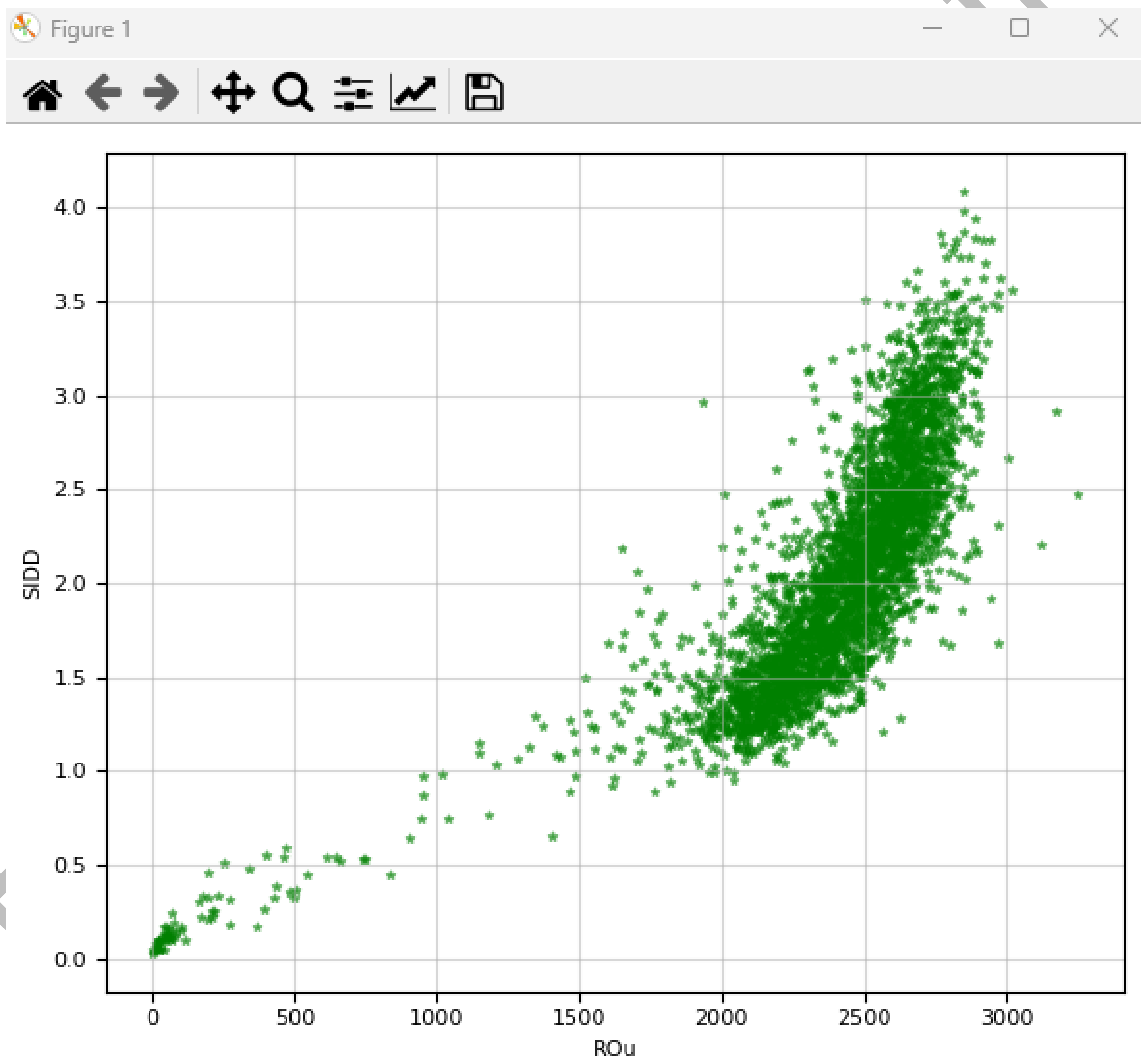

Fig. 30 ROu vs SIDD Scatter Analysis (for example).

6.2. Data Augmentation

After loaded data, user can define new features in console panel, as follows:

```
Console
Copernic System Beta Release
DigWise Technology Co., Ltd. Copyright © 2024
DTCO> print(df)
        X   Y      SIDD  ...  IDS_LVT_P  IDS_ULVT_N  IDS_ULVT_P
LWID                     ...
1      30   1  0.050494  ...    2.77679     8.99698     6.12671
1      31   1  0.214836  ...    3.11498     9.87748     6.85812
1      32   1  0.149110  ...    2.88482     8.28885     5.89864
1      33   1  0.177254  ...    3.05834     9.33211     7.10189
1      34   1  0.123452  ...    2.89711     9.01821     6.52277
...    ..  ..       ...  ...        ...         ...         ...
300    37  65  2.151300  ...    3.53708    10.20130     7.71509
300    38  65  1.996200  ...    3.55534    10.04750     7.40541
300    39  65  1.136370  ...    3.55109    10.90370     8.18230
300    40  65  0.301952  ...    3.36258     9.84973     7.02725
300    41  65  0.108932  ...    3.18385    10.53510     7.64258

[1009500 rows x 16 columns]
DTCO> df['augmented_feature'] = 1
DTCO> print(df)
        X   Y      SIDD  ...  IDS_ULVT_N  IDS_ULVT_P  augmented_feature
LWID                     ...
1      30   1  0.050494  ...     8.99698     6.12671                  1
1      31   1  0.214836  ...     9.87748     6.85812                  1
1      32   1  0.149110  ...     8.28885     5.89864                  1
1      33   1  0.177254  ...     9.33211     7.10189                  1
1      34   1  0.123452  ...     9.01821     6.52277                  1
...    ..  ..       ...  ...         ...         ...                ...
300    37  65  2.151300  ...    10.20130     7.71509                  1
300    38  65  1.996200  ...    10.04750     7.40541                  1
300    39  65  1.136370  ...    10.90370     8.18230                  1
300    40  65  0.301952  ...     9.84973     7.02725                  1
300    41  65  0.108932  ...    10.53510     7.64258                  1

[1009500 rows x 17 columns]
DTCO>
```

Fig. 31 Data Augmentation (for example).

6.3. Save Data

User can save the dataframe as .csv format, the command as follows:

2. Reset index:
    - Use command: `df.reset_index(inplace=True,drop=False)`, and print to check.

```
DTCO> df.reset_index(inplace=True,drop=False)
DTCO> print(df)
          LWID   X   Y  ...  IDS_ULVT_N  IDS_ULVT_P  augmented_feature
0            1  30   1  ...     8.99698     6.12671                  1
1            1  31   1  ...     9.87748     6.85812                  1
2            1  32   1  ...     8.28885     5.89864                  1
3            1  33   1  ...     9.33211     7.10189                  1
4            1  34   1  ...     9.01821     6.52277                  1
...        ...  ..  ..  ...         ...         ...                ...
1009495    300  37  65  ...    10.20130     7.71509                  1
1009496    300  38  65  ...    10.04750     7.40541                  1
1009497    300  39  65  ...    10.90370     8.18230                  1
1009498    300  40  65  ...     9.84973     7.02725                  1
1009499    300  41  65  ...    10.53510     7.64258                  1

[1009500 rows x 18 columns]
```

Fig. 27 Data reset index (for example).

3. Save df as .csv format:
    - Save data by command: `df.to_csv('your path//file_name.csv',index=False)`

```
DTCO> df.to_csv('C://River_TUF F15//Digwise//Work Contents//Copernic_System_Beta//test.csv', index=False)
```

If save successfully, you will find the {name}.csv file saved in your path, and you can reload it to analysis.

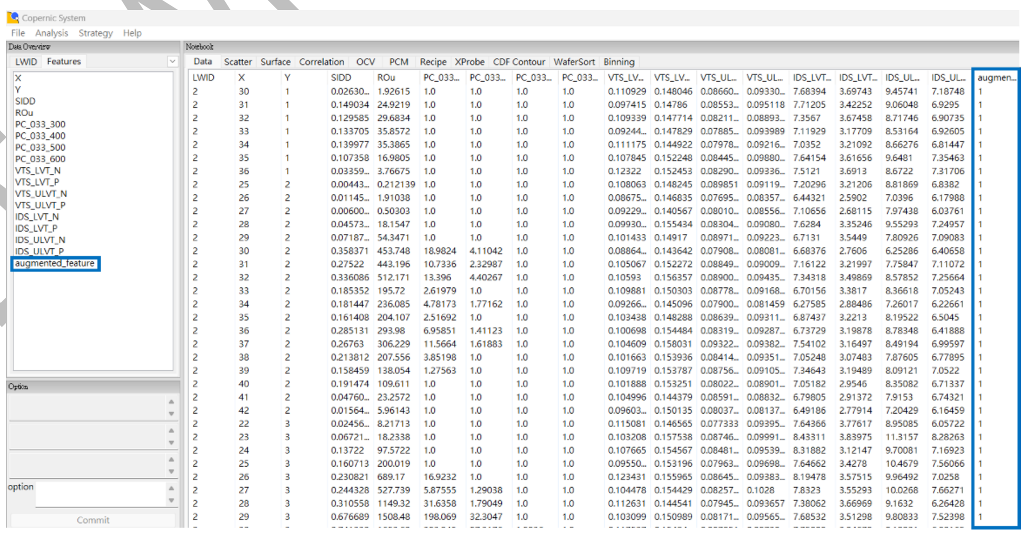

Fig. 32 Augmented data reloaded (for example).

7. API Introduction

This section provides an in-depth introduction to the core API functions used for data analysis. These functions are designed to handle high-dimensional data, perform statistical computations, visualize trends, and offer a flexible and efficient data processing framework. Each API has been optimized to ensure low time complexity and high computational efficiency when working with large datasets. Additionally, the cross dynamic CP-WAT analysis capabilities allow for comprehensive evaluation of multi-variable interactions and trends within the data. Below is a detailed description of each API's main purpose and its application scenarios:

- **Data Loading and Preprocessing**: Supports importing data in various formats (including CSV, XLSX, PKL, etc.) and offers automated cleaning and preprocessing functions. This API is ideal for quickly loading and preparing large datasets for further analysis.
- **Data Visualization**: Provides multiple graphical rendering options such as 2D and 3D scatter plots, heatmaps, etc. It allows users to generate intuitive visual representations quickly, aiding in the analysis of correlations and trends within the data.
- **Statistical Analysis**: Equipped with various statistical models and regression analysis methods, this API allows users to configure parameters for customized analysis, making it suitable for identifying complex data patterns and building predictive models.
- **CP-WAT Cross Probing Analysis**: This specialized API enables in-depth analysis of cross-variable interactions and dynamic changes over time. It integrates various parameters to assess how different factors influence the overall performance and behavior of the system, providing insights into critical dynamics within the dataset.
- **Interpolation and Curve Fitting**: Designed to perform interpolation on irregular datasets, supporting multiple fitting methods (such as linear interpolation and spline interpolation). This ensures smooth and accurate data interpolation results in high-dimensional spaces.

7.1. batchFeature

Purpose: Generates a grid of wafer-level feature visualizations for selected features.

Visualization Types: Supports both 2D contour plots and 3D surface plots.

Comparison: Allows users to visually compare feature distributions across multiple wafers.

Data Filtering: Filters data based on a quantile range to focus on central values.

Parameters:

Parameter	Type	Default	Description
df	DataFrame	Required	The DataFrame containing wafer data, with features like X, Y, and the target feature (feature). Indexed by wafer ID (LWID).
feature	str	'SIDD'	The target feature to be visualized for each wafer (e.g., 'SIDD', 'VTS_LVT_P').
lwidL	list	None	List of wafer IDs (LWID) to include in the visualization. If None, all wafers in the DataFrame are considered.
num	int	Calculated by len	The number of wafers to plot. Defaults to 20 if more than 20 wafers are present, otherwise uses the length of lwidL.
ncol	int	5	The number of columns in the subplot grid. Determines the arrangement of wafer plots.
dtype	str	'2d'	Determines whether the plot is in 2D or 3D mode. Use '2d' for 2D contour plots and '3d' for 3D surface plots.
view	tuple	(30,-35)	A tuple specifying the elevation and azimuthal angles for 3D plots. This adjusts the viewing angle of the 3D surfaces.
zoom	float	1.7	Controls the zoom level for the 3D plot. The higher the value, the closer the view.

Example Code:

```
batchFeature(df,feature='SIDD',lwidL=None,num=None,ncol=None,dtype='2d',
view=(30,-35),zoom=1.7)
```

7.2. report_scatter_grid

Purpose: Generates a high-dimensional scatter plot grid to visualize pairwise relationships between features.

Functionality: Creates a grid of scatter plots comparing pairs of features from specified x-axis and y-axis lists; supports interactive capabilities for users to select specific data points.

Data visualization: Allows visualization of detailed information for a subset of the data based on user selection.

Parameters:

Parameter	Type	Default	Description
dt	Pandas.DataFrame	Required	The input data frame indexed by LWID, containing the metrics/features to be plotted.
fxL	list (optional)	['ROu', 'PC_033_300', 'VTS_LVT_N']	A list of feature names for the x-axis in each scatter plot.
fyL	list (optional)	['SIDD', 'VTS_LVT_P', 'VTS_ULVT_P']	A list of feature names for the y-axis in each scatter plot.
args	dict (optional)	{'figsize': (10,8), 'color': 'blue', ...}	Additional parameters to control the appearance of the scatter plot (detailed in the table below).

args Dictionary Details:

Key	Type	Default	Description
figsize	tuple	(10, 8)	Size of the figure (width, height).
color	str	'blue'	Color of the scatter points.
marker	str	'o'	Marker style for scatter points.
s	int	10	Size of the markers.
alpha	float	0.5	Transparency level of the markers (0.0 to 1.0).
fontsize	int	8	Font size for axis labels.
labelsize	int	8	Font size for tick labels.

Example Code:

```
report_scatter_grid(dt,fxL=['ROu','PC_033_300','VTS_LVT_N'],fyL=['SIDD','VTS_LVT_P','VTS_ULVT_P'],args={'figsize':(10,8),'color':'blue','marker':'o','s':10,'alpha':0.5,'fontsize':8,'labelsize':8})
```

7.3. report_scatter3d_grid

Purpose: Generates a 3D scatter plot grid to visualize relationships between specified features.

Visualization Types: Creates a grid of 3D scatter plots to explore relationships between multiple sets of features.

Configuration Options: Offers customization for plot appearance, view angles, and marker properties to enhance visualization.

Hint: Limit the scatter matrix to less than 10x10.

Parameters:

Parameter	Type	Default	Description
dt	Pandas.DataFrame	Required	The input data frame indexed by LWID, containing the metrics/features to be plotted.
fxL	list (optional)	['X']	List of feature names to use as X-axis in the scatter plots.
fyL	list (optional)	['Y']	List of feature names to use as Y-axis in the scatter plots.
fzL	list (optional)	['ROu', 'SIDD']	List of feature names to use as Z-axis in the scatter plots.
args	dict (optional)	{'figsize': (10,8), 'view': (30, -35), ...}	Additional parameters to control the appearance of the scatter plot (detailed in the table below).

args Dictionary Details:

Key	Type	Default	Description
figsize	tuple	(10, 8)	Size of the figure (width, height).
view	tuple	(30, -35)	Elevation and azimuthal angles for 3D view.
zoom	float	1.0	Zoom level for the plot.
color	str	'blue'	Color of the scatter markers.
marker	str	'o'	Style of the scatter markers.
s	int	10	Size of the scatter markers.
alpha	float	0.5	Transparency level of the markers (0.0 to 1.0).
fontsize	int	8	Font size for axis labels.
labelsize	int	8	Font size for tick labels.

Example Code:

```
report_scatter3d_grid(dt,fxL=['X'],fyL=['Y'],fzL=['ROu','SIDD'],args={'fi
gsize':(10,8),'view':(30,-
35),'zoom':1.0,'color':'blue','marker':'o','s':10,
'alpha':0.5,'fontsize':8,'labelsize':8})
```

7.4. report_surface_grid

Purpose: Generates a grid of 3D surface plots to visualize the relationship between specified features.

Visualization: Creates a grid of 3D surface plots for intuitive visualization of interactions among multiple features over a 2D grid.

Customization: Supports various plot customizations, including viewing angles, zoom levels, and visual styling for smooth surfaces.

Exploration: Helps users effectively explore and understand complex feature interactions in their dataset.

Parameters:

Parameter	Type	Default	Description
dt	Pandas.DataFrame	Required	The input data containing the features to be visualized on a 3D surface grid.
fxL	list (optional)	['X']	List of feature names to use as X-axis in the surface plots.
fyL	list (optional)	['Y']	List of feature names to use as Y-axis in the surface plots.
fzL	list (optional)	['ROu', 'SIDD']	List of feature names to use as Z-axis in the surface plots.
args	dict (optional)	{'figsize': (10,8), 'view': (30, -35), ...}	Additional parameters to control the appearance of the scatter plot (detailed in the table below).

args Dictionary Details:

Key	Type	Default	Description
figsize	tuple	(10, 6)	Size of the figure (width, height).
zoom	float	1.15	Zoom level for the plot.
view	tuple	(70, -50)	Elevation and azimuthal angles for 3D view.
alpha	float	0.5	Transparency level of the markers (0.0 to 1.0).
linewidth	float	0	Width of the surface lines
edgecolors	str	'none'	Color of the edges around the surface elements
fontsize	int	8	Font size for axis labels.
labelsize	int	8	Font size for tick labels.

Example Code:

```
report_surface_grid(dt,fxL='X',fyL='Y',fzL=['ROu','SIDD'],args={'figsize':
(10,6),'zoom':1.15,'view':(70,-50),'alpha':0.5,'linewidth':0,'edgecolors':
'none','fontsize':8,'labelsize':8})
```

7.5. report_correlation

Purpose: Generates a correlation matrix and visualizes pairwise feature correlations.

Visualization: Creates a correlation matrix for a specified list of features and visualizes the pairwise correlations using both scatter plots and histograms. This offers a comprehensive view of feature relationships.

Customization: Supports plot customization, including appearance adjustments, label formatting, and displaying correlation coefficients. These options allow users to tailor the visualization to better analyze feature dependencies.

Exploration: Helps users understand and explore the dependencies and correlations between multiple features in a dataset.

Parameters:

Parameter	Type	Default	Description
dt	Pandas.DataFrame	Required	The input data frame indexed by LWID, containing the metrics/features to be plotted.
itemL	list (optional)	['SIDD', 'ROu', 'VTS_ULVT_N', 'VTS_ULVT_P', 'VTS_LVT_N', 'VTS_LVT_P']	List of feature names to include in the correlation analysis.
args	dict (optional)	{'figsize': (10,8), 's': 15, ...}	Additional parameters to control the appearance of the scatter plot (detailed in the table below).

args Dictionary Details:

Key	Type	Default	Description
figsize	tuple	(10, 8)	Size of the figure (width, height).
s	int	15	Marker size for scatter plots.
bins	int	30	Number of bins for histograms.
color	str	'b'	Color of the scatter points.
alpha	float	0.5	Transparency level for scatter points.
fontsize	int	10	Font size for axis labels.
corrsize	int	12	Font size for correlation coefficients.
corr_x	float	0.1	X-position for the correlation coefficient display.
corr_y	float	0.1	Y-position for the correlation coefficient display.
labelpad	int	12	Padding for axis labels.
x_rotate	int	0	Rotation angle for x-axis labels.
y_rotate	int	90	Rotation angle for y-axis labels.

Example Code:

```
report_correlation(dt,itemL=['SIDD','ROu','VTS_ULVT_N','VTS_ULVT_P','VTS_LVT_N','VTS_LVT_P'],args={'figsize':(10,8),'s':15,'bins':30,'color':'b','alpha':0.5,'fontsize':10,'corrsize':12,'corr_x':0.1,'corr_y':0.1,'labelpad':12,'x_rotate':0,'y_rotate':90})
```

7.6. d2dDerating

Purpose: Performs die-to-die derating analysis for the specified feature across selected wafers.

Analysis: Calculates die-to-die variations and applies derating based on standard deviation (SD) and sigma levels for a given feature across multiple wafers. This analysis helps in identifying performance deviations and ensures that the feature's distribution falls within acceptable limits, accounting for manufacturing variability.

Customization: Supports adjustments for sigma levels and offers control over the analysis boundaries, providing flexibility in determining acceptable performance ranges.

Exploration: Enables users to monitor and assess die-to-die performance variations, ensuring quality control and consistency across wafers in a manufacturing process.

Parameters:

Parameter	Type	Default	Description
df	Pandas.DataFrame	Required	The input dataframe containing feature values for different wafers, indexed by 'LWID'.
lwidL	list (optional)	[]	List of wafer IDs (LWID) to include in the analysis. If empty, the function considers all wafers in the dataframe
feature	str	'ROu'	The target feature to be analyzed for derating.
sd	int	3	Standard deviation range used to filter outliers. Features beyond sd standard deviations from the mean will be excluded.
sigma	float	2.0	The sigma level used for the derating calculation. A higher sigma value results in stricter derating.
detail	int	1	Controls the level of detail in the analysis output. A value of 1 provides basic output, while higher values increase verbosity for debugging or detailed logs.

Example Code:

```
d2dDerating(df,lwid=[],feature='ROu',sd=3,sigma=2.0,detail=1)
```

7.7. pcmDensity

Purpose: Generates a 2D density plot to visualize the distribution of two features across multiple wafers.

Visualization: Creates a 2D density plot for the specified features, offering a clear representation of feature interactions and their distribution. The plot highlights regions of high density, providing insights into feature concentration.

Customization: Supports filtering based on sigma levels or specific percentiles, allowing users to focus on key regions or trends within the data. Additional options for plot appearance and axis labeling are available for enhanced clarity.

Exploration: Helps users identify trends, anomalies, or critical regions within the data, aiding in deeper exploration of feature interactions and distribution patterns across wafers.

Parameters:

Parameter	Type	Default	Description
df	Pandas.DataFrame	Required	The input dataframe containing feature values for different wafers, indexed by 'LWID'.
lwidL	list (optional)	None	List of wafer IDs (LWID) to include in the analysis. If None, all wafers in the dataframe are considered.
fx	str	'ROu'	The feature to be used for the X-axis in the density plot.
fy	str	'SIDD'	The feature to be used for the Y-axis in the density plot.
sigma	float	None	The sigma level used to filter the data. If provided, only data within sigma standard deviations of the mean is plotted.
percentiles	list	[10, 20, 30]	A list of percentiles to highlight specific regions of the density plot.
bins	tuple	(100, 100)	The number of bins for the X and Y axes in the density plot, controlling the resolution of the density estimation.
sub	int	1	Subsampling factor for the data. A value greater than 1 reduces the data size, making the analysis faster.

Example Code:

```
pcmDensity(df,lwidL=None,fx='ROu',fy='SIDD',sigma=None,percentiles=[10,20
,30],bins=(100,100),sub=1)
```

7.8. surfaceContour

Purpose: Generates a surface contour plot to visualize the relationship between two features and a target variable.

Visualization: Creates a 2D contour plot representing the relationship between two independent features (X and Y) and a target feature (Z). The contour lines provide a visual representation of the surface formed by the target feature, illustrating its distribution across the feature space.

Customization: Supports various visual adjustments, including contour levels, color mapping, and axis labels, allowing users to tailor the plot for clearer representation of feature interactions.

Exploration: Enables users to analyze the underlying distribution and interaction between the features, helping to identify key trends or regions of interest in the dataset.

Parameters:

Parameter	Type	Default	Description
df	Pandas.DataFrame	Required	The input dataframe containing the feature values to be plotted.
fx	str	-	The feature to be plotted along the X-axis.
fy	str	-	The feature to be plotted along the Y-axis.
fz	str	-	The target feature to be represented by the contour levels.
fsize	tuple	(6, 5)	The size of the figure (width, height) for the plot.
levels	int	10	The number of contour levels to be plotted, controlling the granularity of the contours.

Example Code:

```
surfaceContour(df,fx,fy,fz,fsize=(6,5),levels=10)
```

7.9. crossProbing

Purpose: Performs cross-feature probing to analyze the correlation between two sets of features across multiple data points.

Visualization: Generates scatter plots to compare two sets of features, enabling users to explore the correlations between different variables. The function can compute and display the mean values for each feature set, providing additional insights into their distribution and relationships.

Customization: Supports customization of point styles, edge colors, and mean lines, allowing users to enhance the visual clarity of the scatter plots for better analysis and understanding.

Exploration: Helps users explore feature correlations across multiple data points, offering a detailed view of how different variables relate and interact with each other.

Parameters:

Parameter	Type	Default	Description
dt	pandas.DataFrame	Required	The input dataframe containing the feature values for cross-probing.
f1	tuple	('AC_uLVT_core', 'SIDD_VDDCPU_V100')	A tuple of feature names for the first set to be plotted.
f1	tuple	('Vtl_N4UL', 'Vtl_P3UL')	A tuple of feature names for the second set to be plotted.
mean	bool	False	If True, the function will plot the mean values of each feature set in addition to individual points.
args	dict (optional)	{'mean_c1': 'blue', 'mean_c2': 'orange', ...}	Additional parameters to control the appearance of the scatter plot (detailed in the table below).

args Dictionary Details:

Key	Type	Default	Description
mean_c1	str	'blue'	Color for the mean point of the first feature set.
mean_c2	str	'orange'	Color for the mean point of the second feature set.
mean_lw	float	3	Line width for the mean markers.
mean_s	int	200	Size of the mean point markers.
point_s	int	10	Size of individual data point markers.
edge_c	str	'purple'	Edge color for the points.
edge_lw	float	4	Line width for the point edges.

Example Code:

```
crossProbing(dt,f1=('AC_uLVT_core','SIDD_VDDCPU_V100'),f2=('Vtl_N4UL','Vt
l_P3UL'),mean=False,args={'mean_c1':'blue','mean_c2':'orange','mean_lw':3
,'mean_s':200,'point_s':10,'edge_c':'purple','edge_lw':4})
```

7.10. CDF Contour

Purpose: Generates a CDF (Cumulative Distribution Function) contour plot to visualize feature distributions across multiple wafers.

Visualization: Creates a contour plot based on the cumulative distribution of two features across specified wafers. The plot highlights specific percentile levels, offering insights into how feature values are distributed within the dataset.

Customization: Optionally, a surface plot can be generated for deeper visualization. The function also supports dropping outliers from the analysis to focus on core data trends.

Exploration: Helps users analyze feature distributions and identify key percentile levels, providing a detailed view of how features vary across wafers.

Parameters:

Parameter	Type	Default	Description
df	pandas.DataFrame	Required	The input dataframe containing the feature values for different wafers, indexed by 'LWID'.
lwidL	list	-	List of wafer IDs (LWID) to include in the CDF contour analysis.
fx	str	'ROu'	The feature to be used for the X-axis in the contour plot (optional).
fy	str	'SIDD'	The feature to be used for the Y-axis in the contour plot (optional).
percentiles	list	[5, 50, 95]	A list of percentiles to be displayed in the contour plot (optional).
surface	bool	False	If True, generates a 3D surface plot of the CDF instead of a 2D contour plot (optional).
drop	bool	False	If True, outliers beyond the specified percentiles are dropped from the analysis (optional).
flag	bool	True	If True, includes additional flags or markers in the plot to indicate special conditions or outliers.

Example Code:

```
pcmCDFContour(df,lwidL,fx='ROu',fy='SIDD',percentiles=[5,50,95],surface=F
alse,drop=False,flag=True)
```

7.11. waferSort

Purpose: Sorts wafers based on the specified features and returns the sorted DataFrame for further analysis.

Functionality: Sorts the wafers in the dataset using the values of one or more specified features. This allows users to rank the wafers based on chosen metrics, enabling a structured view of wafer performance.

Customization: Users can define multiple features to guide the sorting process and optionally limit the number of wafers considered in the sorting.

Exploration: The sorted DataFrame can be used for further analysis or visualization, helping users evaluate wafer performance across different metrics.

Parameters:

Parameter	Type	Default	Description
df	pandas.DataFrame	Required	The input dataframe containing the wafer data, with each row representing a wafer.
itemL	list	['ROu', 'SIDD']	A list of feature names to be used for sorting the wafers (optional).
nsize	int	None	The number of wafers to include in the sorted dataframe. If None, all wafers are included (optional).

Example Code:

```
waferSort(df,itemL=['ROu','SIDD'],nsize=None)
```

7.12. pcmBinning

Purpose: Performs binning on wafer data based on specified features to analyze distributions across defined bins.

Functionality: Segments the wafer data into defined bins for two selected features, helping to visualize and analyze the density or distribution of data points within each bin. This facilitates a more detailed and granular analysis of feature interactions.

Customization: Users can customize the number of bins and their ranges, and apply optional outlier handling using sigma levels to focus on core data trends.

Exploration: Enables users to break down and explore feature distributions in a structured manner, aiding in the identification of patterns or anomalies within the dataset.

Parameters:

Parameter	Type	Default	Description
df	pandas.DataFrame	Required	The input dataframe containing the feature values for different wafers, indexed by LWID.
lwidL	list	None	List of wafer IDs (LWID) to include in the binning analysis. If None, all wafers are considered.
fx	str	'ROu'	The feature to be used for the X-axis in the binning analysis.
fy	str	'SIDD'	The feature to be used for the Y-axis in the binning analysis.
sigma	float	None	The number of standard deviations for identifying and handling outliers in the data.
bins	tuple	(6, 6)	A tuple defining the number of bins along the X and Y axes.
rangeL	tuple	None	A tuple specifying the lower and upper limits for the binning ranges along the X and Y axes. If None, limits are based on data range.

Example Code:

```
pcmBinning(df,lwidL=None,fx='ROu',fy='SIDD',sigma=None,bins=(6,6),rangeL=
None)
```

7.13. featureSurface

Purpose: Generates a 3D surface plot to visualize the distribution of a specified feature across wafers.

Functionality: Creates a surface plot that represents the values of a selected feature over the wafer data, providing a visual representation of feature distribution. Users can filter the data based on wafer IDs and handle outliers using a specified sigma value.

Customization: Allows customization of viewing angles and transparency levels to enhance visualization and clarity.

Exploration: Enables users to analyze the distribution of features across wafers in a three-dimensional context, facilitating a deeper understanding of the dataset.

Parameters:

Parameter	Type	Default	Description
df	pandas.DataFrame	Required	The input dataframe containing feature values for different wafers, indexed by LWID.
lwidL	list	None	List of wafer IDs (LWID) to include in the surface plot. If None, all wafers in the DataFrame are considered.
feature	str	'SIDD'	The feature to be visualized in the surface plot.
sigma	float	None	The number of standard deviations for identifying and handling outliers in the data.
xyL	list	None	A list specifying the X and Y features to be used for the surface plot. Defaults to the first two features in the dataframe.
alpha	float	1	Controls the transparency of the surface plot. A value of 1 is fully opaque, while 0 is fully transparent.
view	tuple	(70, 300)	A tuple specifying the elevation and azimuthal angles for the 3D view.

Example Code:

```
featureSurface(self,df,lwid=None,feature='SIDD',sigma=None,xyL=None,alpha
=1,view=(70,300))
```

8. Exporting Results

To export your analysis results, follow these steps:

1. Access the Export Option:
    - Navigate to the "**File > Export Results"** menu option to begin saving your analysis.
2. Specify Save Location and Format:
    - In the file dialog, choose the desired location and provide a filename. Ensure the format is set to HTML for web-compatible output.
3. Locate the Suggestions File:
    - Once saved, the file (e.g., **suggestions.html**) will be available in your selected directory, ready for review or sharing.

9. Tips and Tricks

1. Batch Analysis Performance:
    - For batch analyses, it is recommended to select more than 10 wafers to ensure optimal performance and more reliable trend identification.

For further assistance or to report issues, please contact our support team.