DigWise Technology Corporation, LTD.

libMetric™ User Guide

MLD
2024/12/20

Table of Contents

1. Introduction 2
2. Core Features 2
3. Getting Started 2
   3.1. Loading Data 2
   3.2. User Interface Overview 3
   3.3. Basic operations 4
4. Performing Analysis 6
   4.1. PPA (Power, Performance, Area) Analysis 6
   4.2. Timing Balance Analysis 7
   4.3. Timing Constraint Analysis 8
   4.4. Delay Insertion Analysis 9
   4.5. Cell Metric Analysis 10
5. Advanced Application 11
   5.1. Execute custom scripts for specialized analyses 11
   5.2. Data Augmentation 12
   5.3. Custom Analysis 13
6. API Introduction 14
   6.1. plot_cell_lsc_surface 15
   6.2. cellMetricConstraint 16
   6.3. batchMetricPPA 17
   6.4. batchMetricTimingBalance 18
   6.5. batchMetricConstraint 19
   6.6. batchDelayInsertion 20

1. Introduction

Library Metric is an advanced EDA tool designed for efficient analysis and comparison of standard cell timing libraries. It offers powerful features for extracting critical metrics, facilitating interactive queries, and performing batch comparisons and trend analyses across multiple PVT corners and device conditions.

2. Core Features

- Regression mode extraction of important metrics from hundreds of standard cell timing libraries
- Consolidated database for convenient interactive querying
- Batch execution of comparisons and trend analyses across multiple cells
- Intuitive graphical user interface for easy navigation and visualization

3. Getting Started

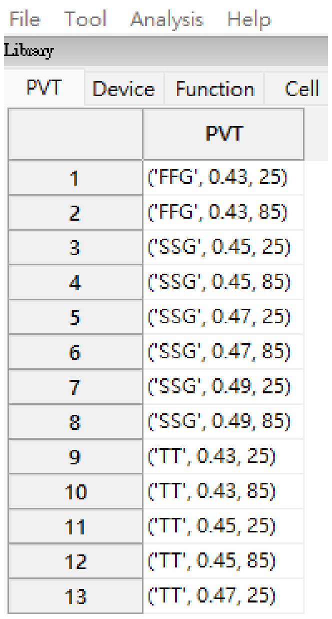
3.1. Loading Data

Before performing Lib to metric, you need to select the lib file to be used as the basegate for generating the INFO file. Then need to convert and merge multiple PVT library files. To begin using Library Metric, you need to load your standard cell timing library data.

1. Create INFO.json File:
    - Click on **"Tool"** in the top menu, then select **"Create IFNO file"** from the dropdown menu.
    - Select the lib file to be used as the basegate for generating the INFO file. **Attention**: It is crucial to use libraries containing both **NAND** and **DF** for generating the INFO file.
2. Convert and Merge Multiple PVT Library Files:
    - Click on **"Tool"** in the top menu, then select **"Lib to Metric"** from the dropdown menu.

- Select the INFO.json file to obtain the reference parameters.
- Select the library files that you want to convert and merge.

3. Loading Data:
    - Click on **"File"** in the top menu, then select **"Load Metric"** from the dropdown menu.
    - Select the appropriate file(s) in the file dialog. **Library** will update to display the imported information, as illustrated in Fig. 1.



	PVT
1	('FFG', 0.43, 25)
2	('FFG', 0.43, 85)
3	('SSG', 0.45, 25)
4	('SSG', 0.45, 85)
5	('SSG', 0.47, 25)
6	('SSG', 0.47, 85)
7	('SSG', 0.49, 25)
8	('SSG', 0.49, 85)
9	('TT', 0.43, 25)
10	('TT', 0.43, 85)
11	('TT', 0.45, 25)
12	('TT', 0.45, 85)
13	('TT', 0.47, 25)

Fig. 1 Library display imported Metric information.

3.2. User Interface Overview

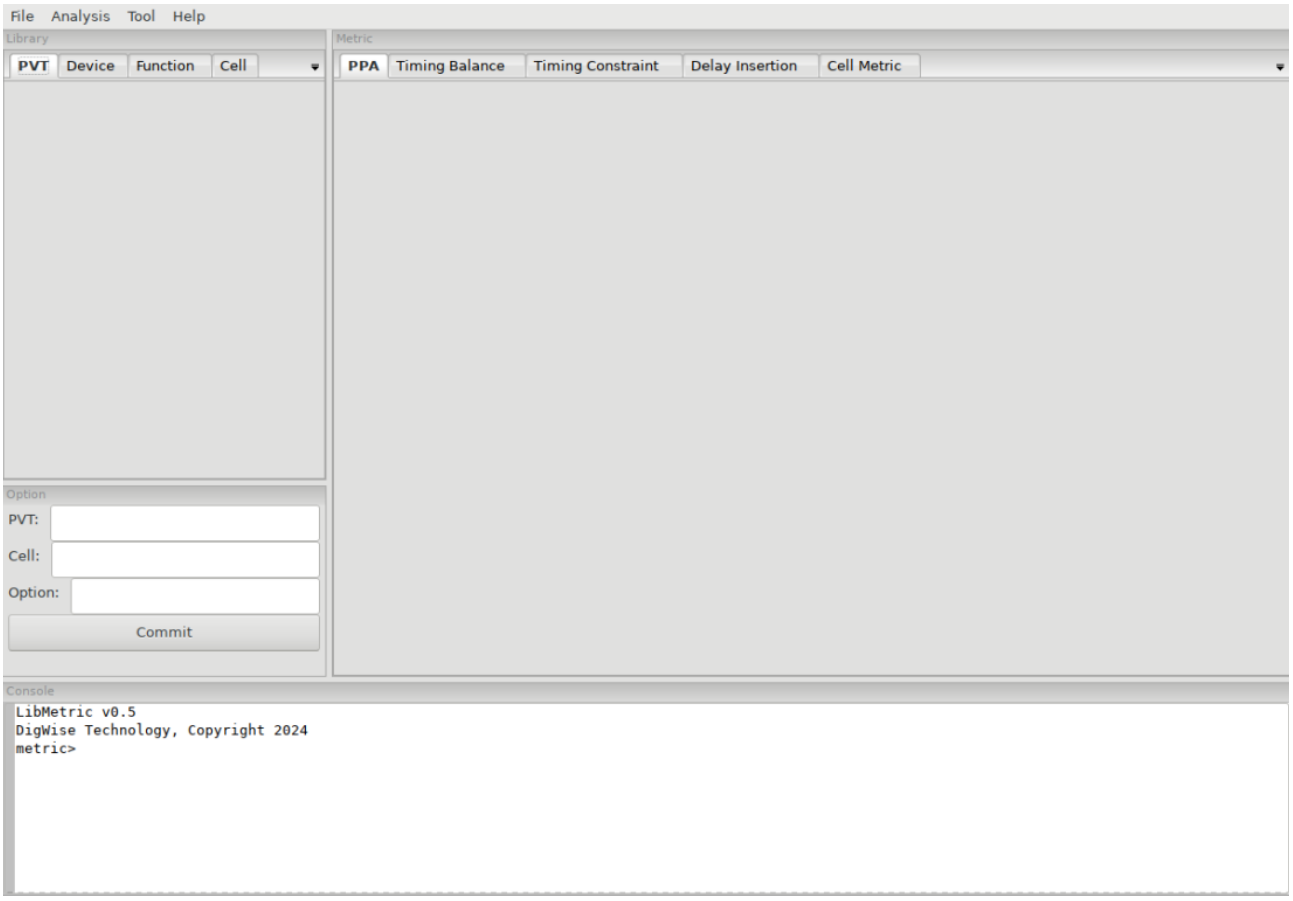
This section will sequentially introduce the basic operations for data selection within each of the four main panels of the main window.

- *Library Panel*:
  Displays PVT (Process, Voltage, Temperature) conditions, devices, functions, and cells.
- *Option Panel*:
  Allows you to set specific parameters for analysis.
- *Metric Panel*:
  Shows various analysis results and visualizations.
- *Console Panel*:
  Provides a Python console for advanced operations.

3.3. Basic operations

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

- *Library Panel*:
  In the Library panel, you can perform single selection using **"left click"** on PVT, Device, Function, or Cell. Additionally, multi-selection is supported using **"Ctrl + left click"**, **"Shift + left click"**, or **"dragging"**. After making a selection, **"right click"** on a column or row and select **"Apply Filter"** to lock in the selected items.
- *Option Panel*:
  In the Option panel, users can enter the desired PVT and cell name in the PVT and Cell text boxes. For other specific parameters (e.g., the target delay for Delay Insertion), they can be entered in the corresponding text box in the Option panel. Once the inputs are complete, left click **"Commit"** to proceed.
- *Metric Panel*:
  In the Metric panel, for PPA, Timing Balance, Timing Constraint, and Delay Insertion, users can **"left click"** on points of interest to further inspect snapshots of the corresponding cell metrics. In the Cell Metric section, columns or rows can be selected using **"Ctrl + left click"**, **"Shift + left click"**, or **"dragging"**. By **"right clicking"** on a column or row, users can **"left click"** on the 2D, 3D, or group options to conduct further analysis.
- *Console Panel*:
  Provides a Python console for advanced operations.



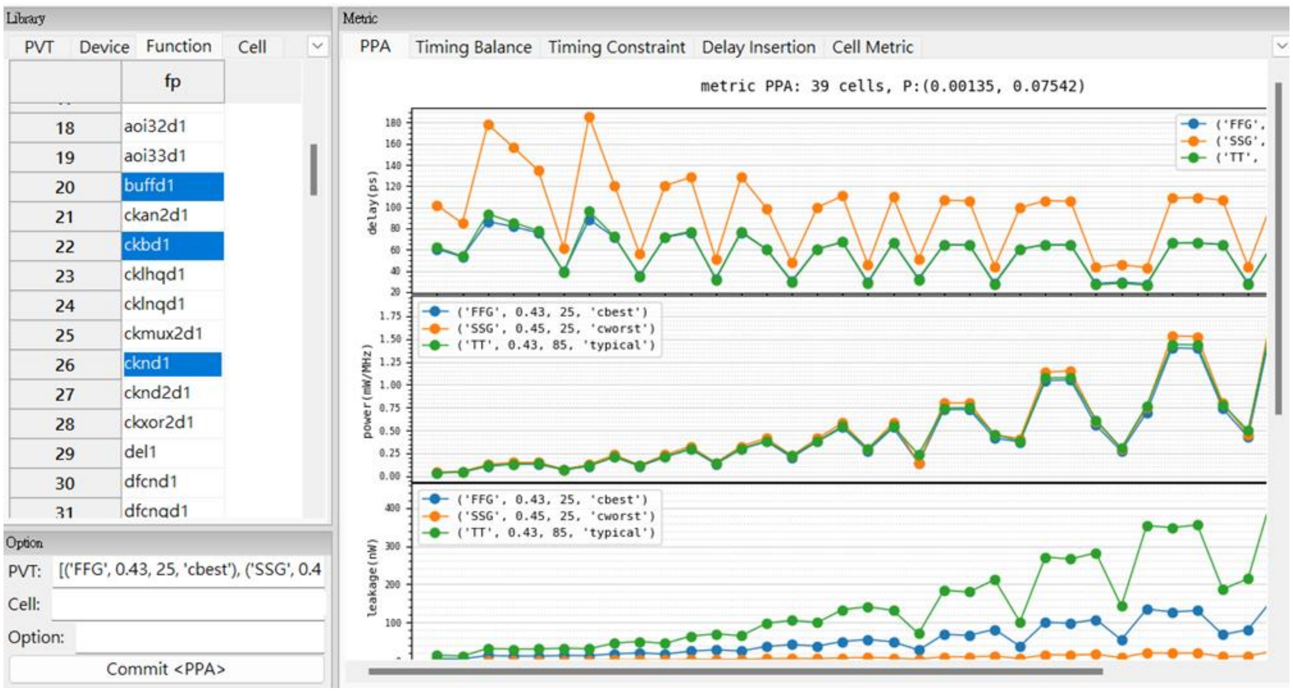
```
LibMetric v0.5
DigWise Technology, Copyright 2024
metric>
```

Fig. 2 libMetric^{TM} main window.

4. Performing Analysis

4.1. PPA (Power, Performance, Area) Analysis

1. Select PVT, Device, Function and Cell:
   - In the **Library Panel**, choose the PVT, Device, Function and Cells you wish to analyze.
2. Access the PPA Tab:
   - Navigate to the **"PPA"** tab located within the **Metric 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 **"Commit <PPA>"** button to execute the plot analysis, which will produce the analysis results for review.



	fp
18	aoi32d1
19	aoi33d1
20	buffd1
21	ckan2d1
22	ckbd1
23	cklhqd1
24	cklnqd1
25	ckmux2d1
26	cknd1
27	cknd2d1
28	ckxor2d1
29	del1
30	dfcnd1
31	dfcnqd1

Option

PVT: [('FFG', 0.43, 25, 'cbest'), ('SSG', 0.4

Cell:

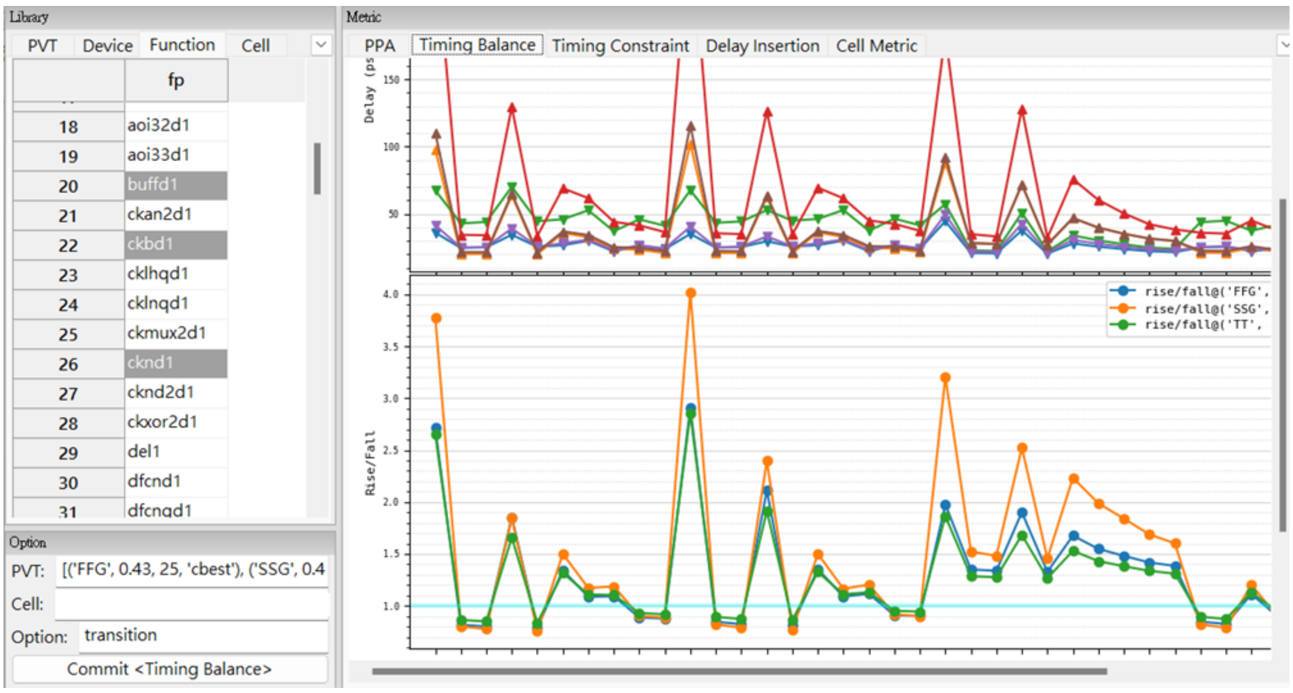
Option:

metric PPA: 39 cells, P:(0.00135, 0.07542)

Fig. 3 PPA Analysis (for example).

4.2. Timing Balance Analysis

1. Select PVT, Device, Function and Cell:
   - In the **Library Panel**, choose the PVT, Device, Function and Cells you wish to analyze.
2. Access the Timing Balance Tab:
   - Navigate to the **"Timing Balance"** tab located within the **Metric 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 **" Commit <Timing Balance>"** button to execute the plot analysis, which will produce the analysis results for review.



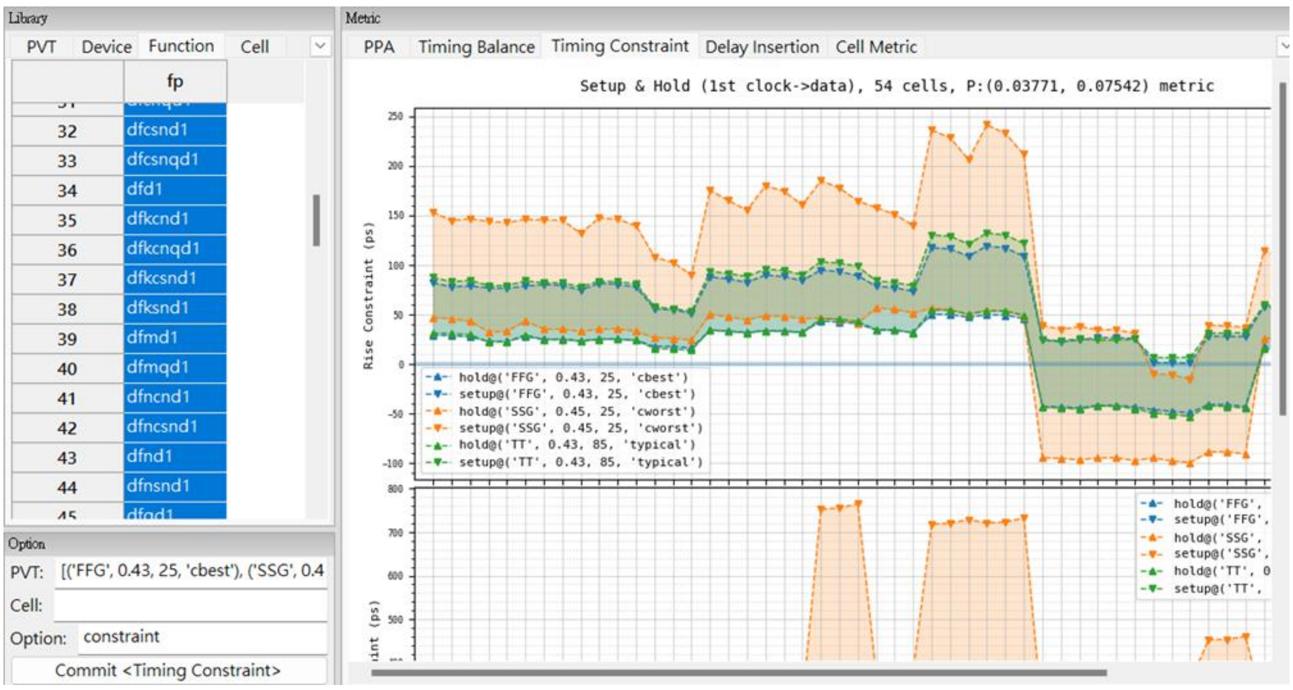
	fp
18	aoi32d1
19	aoi33d1
20	buffd1
21	ckan2d1
22	ckbd1
23	cklhqd1
24	cklnqd1
25	ckmux2d1
26	cknd1
27	cknd2d1
28	ckxor2d1
29	del1
30	dfcnd1
31	dfcnqd1

Option
PVT: [('FFG', 0.43, 25, 'cbest'), ('SSG', 0.4
Cell:
Option: transition

Fig. 4 Timing Balance Analysis (for example).

4.3. Timing Constraint Analysis

1. Select PVT, Device, Function and Cell:
   - In the **Library Panel**, choose the PVT, Device, Function and Cells you wish to analyze.
2. Access the Timing Constraint Tab:
   - Navigate to the **"Timing Constraint"** tab located within the **Metric 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 **" Commit <Timing Constraint>"** button to execute the plot analysis, which will produce the analysis results for review.


Library
PVT
Device
Function
Cell
Metric
PPA
Timing Balance
Timing Constraint
Delay Insertion
Cell Metric
Setup & Hold (1st clock->data), 54 cells, P:(0.03771, 0.07542) metric
Rise Constraint (ps)
250
200
150
100
50
0
-50
-100
800
700
600
500
hold@('FFG', 0.43, 25, 'cbest')
setup@('FFG', 0.43, 25, 'cbest')
hold@('SSG', 0.45, 25, 'cworst')
setup@('SSG', 0.45, 25, 'cworst')
hold@('TT', 0.43, 85, 'typical')
setup@('TT', 0.43, 85, 'typical')
Commit <Timing Constraint>

	fp
32	dfcsnd1
33	dfcsnqd1
34	dfd1
35	dfkcnd1
36	dfkcnqd1
37	dfkcsnd1
38	dfksnd1
39	dfmd1
40	dfmqd1
41	dfncnd1
42	dfncsnd1
43	dfnd1
44	dfnsnd1

Option

PVT: [('FFG', 0.43, 25, 'cbest'), ('SSG', 0.4

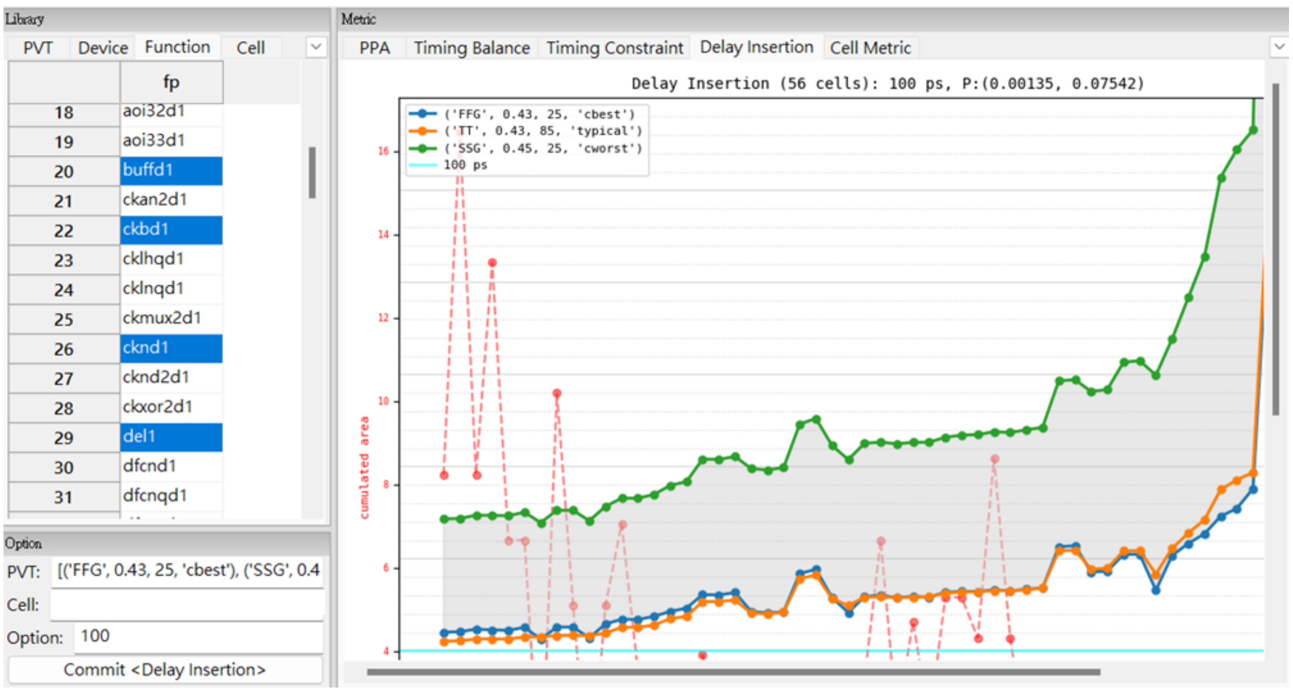
Cell:

Option: constraint

Fig. 5 Timing Constraint Analysis (for example).

4.4. Delay Insertion Analysis

1. Select PVT, Device, Function and Cell:
   - In the **Library Panel**, choose the PVT, Device, Function and Cells you wish to analyze.
2. Access the Delay Insertion Tab
   - Navigate to the **"Delay Insertion"** tab located within the **Metric 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 **" Commit <Delay Insertion>"** button to execute the Delay analysis, which will produce the analysis results for review.



	fp
18	aoi32d1
19	aoi33d1
20	buffd1
21	ckan2d1
22	ckbd1
23	cklhqd1
24	cklnqd1
25	ckmux2d1
26	cknd1
27	cknd2d1
28	ckxor2d1
29	del1
30	dfcnd1
31	dfcnqd1

PVT: [('FFG', 0.43, 25, 'cbest'), ('SSG', 0.4

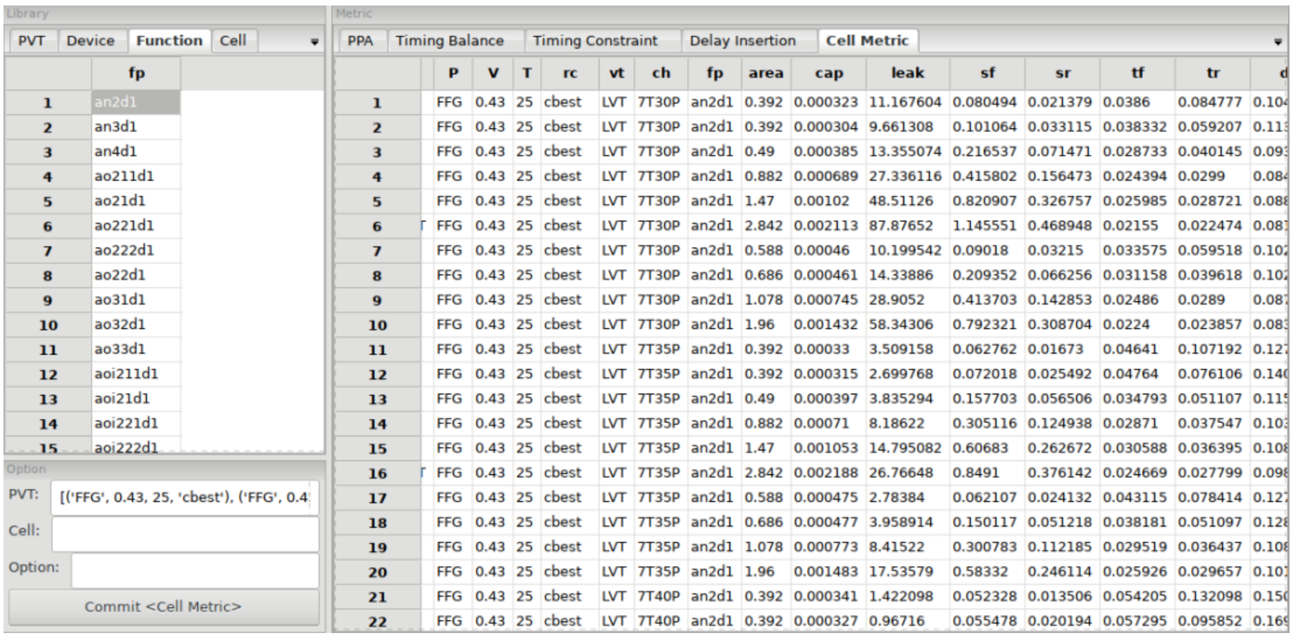
Cell:

Option: 100

Fig. 6 Delay Insertion Analysis (for example).

4.5. Cell Metric Analysis

1. Select PVT, Device, Function and Cell:
   - In the **Library Panel**, choose the PVT, Device, Function and Cells you wish to analyze.
2. Access the Cell Metric Tab:
   - Navigate to the **"Cell Metric"** tab located within the **Metric 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. Show the Metric:
   - Click the **"Commit <Cell Metric>"** button to execute the Metric, which will produce the metric results for review.
5. Generate Results:
   - Use right-click options on the grid for additional visualization options.



	fp
1	an2d1
2	an3d1
3	an4d1
4	ao211d1
5	ao21d1
6	ao221d1
7	ao222d1
8	ao22d1
9	ao31d1
10	ao32d1
11	ao33d1
12	aoi211d1
13	aoi21d1
14	aoi221d1
15	aoi222d1

PVT: [('FFG', 0.43, 25, 'cbest'), ('FFG', 0.4

Cell:

Option:

	P	V	T	rc	vt	ch	fp	area	cap	leak	sf	sr	tf	tr	d
1	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.392	0.000323	11.167604	0.080494	0.021379	0.0386	0.084777	0.104
2	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.392	0.000304	9.661308	0.101064	0.033115	0.038332	0.059207	0.11
3	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.49	0.000385	13.355074	0.216537	0.071471	0.028733	0.040145	0.09
4	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.882	0.000689	27.336116	0.415802	0.156473	0.024394	0.0299	0.084
5	FFG	0.43	25	cbest	LVT	7T30P	an2d1	1.47	0.00102	48.51126	0.820907	0.326757	0.025985	0.028721	0.08
6	FFG	0.43	25	cbest	LVT	7T30P	an2d1	2.842	0.002113	87.87652	1.145551	0.468948	0.02155	0.022474	0.08
7	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.588	0.00046	10.199542	0.09018	0.03215	0.033575	0.059518	0.10
8	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.686	0.000461	14.33886	0.209352	0.066256	0.031158	0.039618	0.10
9	FFG	0.43	25	cbest	LVT	7T30P	an2d1	1.078	0.000745	28.9052	0.413703	0.142853	0.02486	0.0289	0.08
10	FFG	0.43	25	cbest	LVT	7T30P	an2d1	1.96	0.001432	58.34306	0.792321	0.308704	0.0224	0.023857	0.08
11	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.392	0.00033	3.509158	0.062762	0.01673	0.04641	0.107192	0.12
12	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.392	0.000315	2.699768	0.072018	0.025492	0.04764	0.076106	0.14
13	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.49	0.000397	3.835294	0.157703	0.056506	0.034793	0.051107	0.11
14	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.882	0.00071	8.18622	0.305116	0.124938	0.02871	0.037547	0.10
15	FFG	0.43	25	cbest	LVT	7T35P	an2d1	1.47	0.001053	14.795082	0.60683	0.262672	0.030588	0.036395	0.10
16	FFG	0.43	25	cbest	LVT	7T35P	an2d1	2.842	0.002188	26.76648	0.8491	0.376142	0.024669	0.027799	0.09
17	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.588	0.000475	2.78384	0.062107	0.024132	0.043115	0.078414	0.12
18	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.686	0.000477	3.958914	0.150117	0.051218	0.038181	0.051097	0.12
19	FFG	0.43	25	cbest	LVT	7T35P	an2d1	1.078	0.000773	8.41522	0.300783	0.112185	0.029519	0.036437	0.10
20	FFG	0.43	25	cbest	LVT	7T35P	an2d1	1.96	0.001483	17.53579	0.58332	0.246114	0.025926	0.029657	0.10
21	FFG	0.43	25	cbest	LVT	7T40P	an2d1	0.392	0.000341	1.422098	0.052328	0.013506	0.054205	0.132098	0.15
22	FFG	0.43	25	cbest	LVT	7T40P	an2d1	0.392	0.000327	0.96716	0.055478	0.020194	0.057295	0.095852	0.16

Fig. 7 Show the metric (for example).

5. Advanced Application

The Console Panel provides a Python interface for advanced operations. The following will introduce the main usage and features of the Console.

5.1. Execute custom scripts for specialized analyses

Example Code:

```
batchDelayInsertion(dtco.df,cellL=['cell_name_1','cell_name_2','cell_name_3','cell_name_4','cell_name_5'],pvtL=[('FFG',0.43,25,'cbest'),('FFG',0.43,85,'cbest'),('SSG',0.45,25,'cworst')],xy=dtco.param['xy'],xyrange=dtco.param['xyrange'],weight=0.8,targetD=100)
```

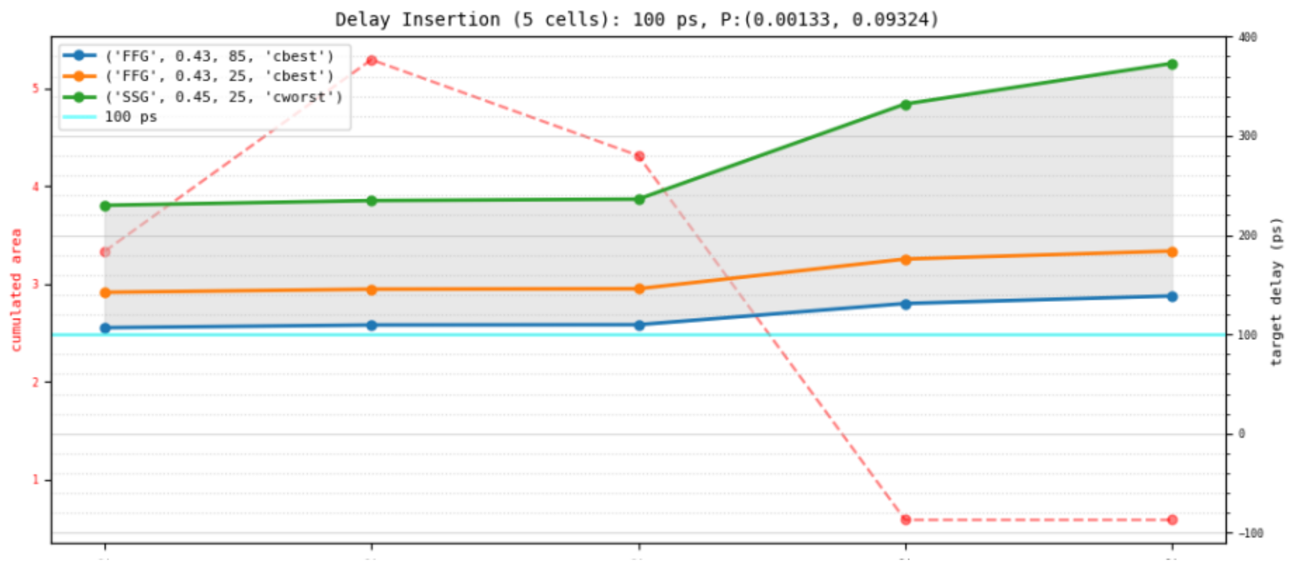
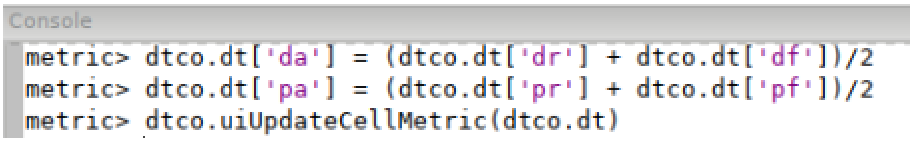


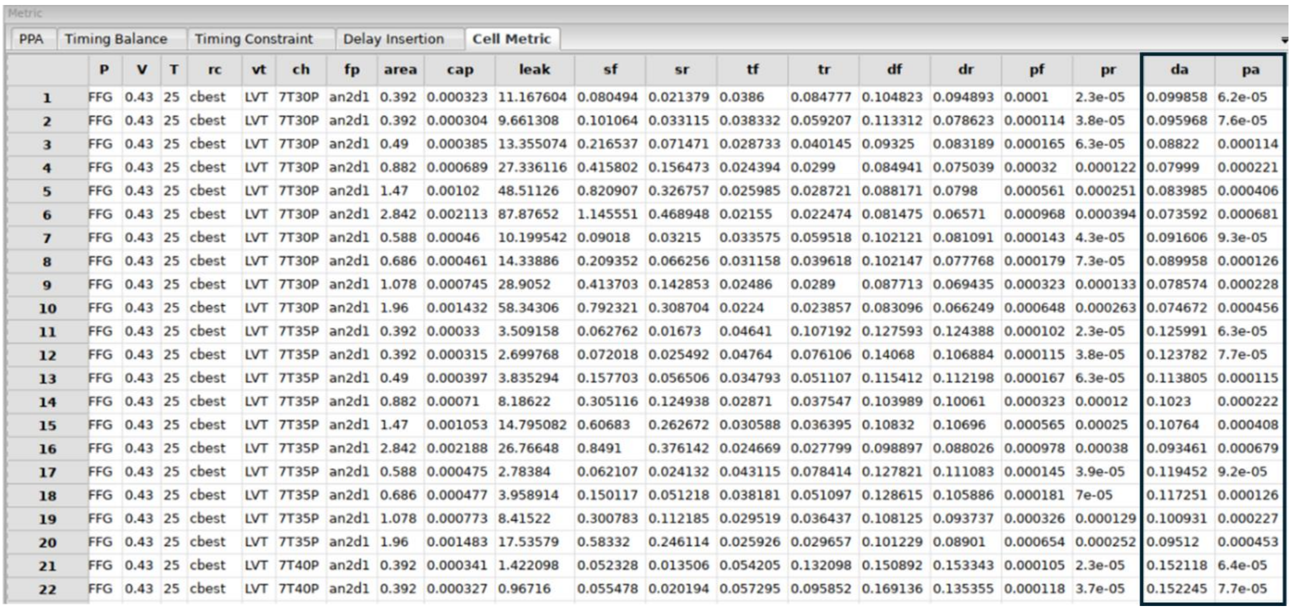
Fig. 8 Delay Insertion Analysis (for example).

5.2. Data Augmentation

User can define new features in console panel, as follows:



```
metric> dtco.dt['da'] = (dtco.dt['dr'] + dtco.dt['df'])/2
metric> dtco.dt['pa'] = (dtco.dt['pr'] + dtco.dt['pf'])/2
metric> dtco.uiUpdateCellMetric(dtco.dt)
```



	P	V	T	rc	vt	ch	fp	area	cap	leak	sf	sr	tf	tr	df	dr	pf	pr	da	pa
1	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.392	0.000323	11.167604	0.080494	0.021379	0.0386	0.084777	0.104823	0.094893	0.0001	2.3e-05	0.099858	6.2e-05
2	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.392	0.000304	9.661308	0.101064	0.033115	0.038332	0.059207	0.113312	0.078623	0.000114	3.8e-05	0.095968	7.6e-05
3	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.49	0.000385	13.355074	0.216537	0.071471	0.028733	0.040145	0.09325	0.083189	0.000165	6.3e-05	0.08822	0.000114
4	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.882	0.000689	27.336116	0.415802	0.156473	0.024394	0.0299	0.084941	0.075039	0.00032	0.000122	0.07999	0.000221
5	FFG	0.43	25	cbest	LVT	7T30P	an2d1	1.47	0.00102	48.51126	0.820907	0.326757	0.025985	0.028721	0.088171	0.0798	0.000561	0.000251	0.083985	0.000406
6	FFG	0.43	25	cbest	LVT	7T30P	an2d1	2.842	0.002113	87.87652	1.145551	0.468948	0.02155	0.022474	0.081475	0.06571	0.000968	0.000394	0.073592	0.000681
7	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.588	0.00046	10.199542	0.09018	0.03215	0.033575	0.059518	0.102121	0.081091	0.000143	4.3e-05	0.091606	9.3e-05
8	FFG	0.43	25	cbest	LVT	7T30P	an2d1	0.686	0.000461	14.33886	0.209352	0.066256	0.031158	0.039618	0.102147	0.077768	0.000179	7.3e-05	0.089958	0.000126
9	FFG	0.43	25	cbest	LVT	7T30P	an2d1	1.078	0.000745	28.9052	0.413703	0.142853	0.02486	0.0289	0.087713	0.069435	0.000323	0.000133	0.078574	0.000228
10	FFG	0.43	25	cbest	LVT	7T30P	an2d1	1.96	0.001432	58.34306	0.792321	0.308704	0.0224	0.023857	0.083096	0.066249	0.000648	0.000263	0.074672	0.000456
11	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.392	0.00033	3.509158	0.062762	0.01673	0.04641	0.107192	0.127593	0.124388	0.000102	2.3e-05	0.125991	6.3e-05
12	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.392	0.000315	2.699768	0.072018	0.025492	0.04764	0.076106	0.14068	0.106884	0.000115	3.8e-05	0.123782	7.7e-05
13	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.49	0.000397	3.835294	0.157703	0.056506	0.034793	0.051107	0.115412	0.112198	0.000167	6.3e-05	0.113805	0.000115
14	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.882	0.00071	8.18622	0.305116	0.124938	0.02871	0.037547	0.103989	0.10061	0.000323	0.00012	0.1023	0.000222
15	FFG	0.43	25	cbest	LVT	7T35P	an2d1	1.47	0.001053	14.795082	0.60683	0.262672	0.030588	0.036395	0.10832	0.10696	0.000565	0.00025	0.10764	0.000408
16	FFG	0.43	25	cbest	LVT	7T35P	an2d1	2.842	0.002188	26.76648	0.8491	0.376142	0.024669	0.027799	0.098897	0.088026	0.000978	0.00038	0.093461	0.000679
17	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.588	0.000475	2.78384	0.062107	0.024132	0.043115	0.078414	0.127821	0.111083	0.000145	3.9e-05	0.119452	9.2e-05
18	FFG	0.43	25	cbest	LVT	7T35P	an2d1	0.686	0.000477	3.958914	0.150117	0.051218	0.038181	0.051097	0.128615	0.105886	0.000181	7e-05	0.117251	0.000126
19	FFG	0.43	25	cbest	LVT	7T35P	an2d1	1.078	0.000773	8.41522	0.300783	0.112185	0.029519	0.036437	0.108125	0.093737	0.000326	0.000129	0.100931	0.000227
20	FFG	0.43	25	cbest	LVT	7T35P	an2d1	1.96	0.001483	17.53579	0.58332	0.246114	0.025926	0.029657	0.101229	0.08901	0.000654	0.000252	0.09512	0.000453
21	FFG	0.43	25	cbest	LVT	7T40P	an2d1	0.392	0.000341	1.422098	0.052328	0.013506	0.054205	0.132098	0.150892	0.153343	0.000105	2.3e-05	0.152118	6.4e-05
22	FFG	0.43	25	cbest	LVT	7T40P	an2d1	0.392	0.000327	0.96716	0.055478	0.020194	0.057295	0.095852	0.169136	0.135355	0.000118	3.7e-05	0.152245	7.7e-05

Fig. 9 Data Augmentation (for example).

5.3. Custom Analysis

Users can perform data analysis in the Console Panel using Python code based on their specific needs, enabling analysis methods beyond the provided APIs. For example, as shown in Fig. 10, the trends of power and performance versus gate length across different PVT conditions are illustrated.

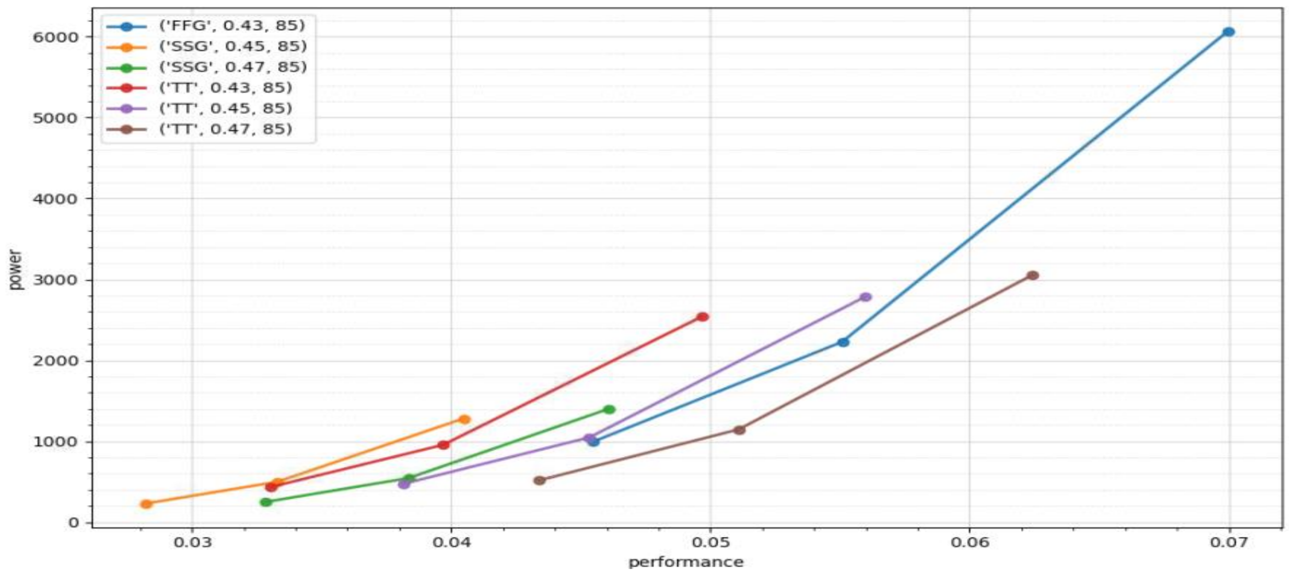


Fig. 10 Custom Analysis (Power vs Performance).

6. API Introduction

This chapter provides an in-depth introduction to the core API functions used for data analysis. These functions are designed to handle Library data merging, Cell information comparison, and data visualization, offering a flexible and efficient data processing framework. Each API is optimized to ensure low time complexity and high computational efficiency when processing large datasets. Additionally, the Lib Metric Extraction & Merge function allows for data extraction from Libraries under different PVT conditions, enabling interactive analysis and providing a more streamlined and convenient comprehensive evaluation. Below is a detailed explanation of each API's primary use and its application scenarios:

- **Data Extraction and Merge API**: Supports importing Libraries in various formats (including .lib, .json) and offers feature extraction and merging capabilities. This API is ideal for integrating Library data under multiple PVT conditions for further analysis.
- **Data Visualization API**: Provides various graphical rendering options such as 2D and 3D scatter plots, multiple line charts, and more. Users can make Cell selection decisions based on the visualization content.
- **Statistical Analysis API**: Equipped with a variety of algorithmic analysis methods, it allows users to configure parameters according to their specific needs, making it suitable for algorithmic ranking of numerous Cells.
- **Cross-PVT Analysis API**: This specialized API enables interactive comparison of multiple Cells under different PVT conditions. It helps users assess the performance trend of Cells under dynamic PVT conditions, aiding in design-related Cell selection decisions.

6.1. plot_cell_lsc_surface

Purpose: Visualize metric surface trends at the cell level.

Visualization Types: 3D surface plots.

Comparison: Allows users to intuitively compare the impact of transition and load on delay across multiple PVTs.

Parameters:

Parameter	Type	Default	Description
df	DataFrame	Required	Metric dataframe indexed with (cell,P,V,T)
cell	str	Required	The target cell name for capturing the timing metric.
pvt	tuple	Required	Input PVT information as needed (supports multiple PVTs).
regex	str	'cell'	Regular expression used for timing arc filtering.
xy	tuple	From INFO.json	The number of columns in the subplot grid. Determines the arrangement of wafer plots.
xyrange	Str	From INFO.json	Referenced index range for LS surface regression, (load,tran) for delay, transition & power, (clock,data) for constraint.

Example Code:

```
plot_cell_lsc_surface(dtco.df,cell='cell_name_1',pvt=('FFG',0.43,25,'cbest'),regex='cell_'
,xy=dtco.param['xy'],xyrange=dtco.param['xyrange'])
```

6.2. cellMetricConstraint

Purpose: Interpolates cell metrics using Least Squares (LS) regression for a given (load, tran) condition, providing insight into the impact of different conditions on delay, transition, and power metrics.

Visualization Types: Generates customized plots to visualize metric surface trends for specific cells under multiple conditions.

Comparison: Allows users to intuitively compare the influence of transition and load on delay across different PVTs. It also supports the visualization of constraint-based metrics such as clock and data timing arcs.

Data Filtering: Filters timing arcs using regular expressions to ensure only relevant data points are included for analysis. This optimizes regression accuracy and visualization clarity.

Parameters:

Parameter	Type	Default	Description
dt	Pandas.DataFrame	Required	DataFrame containing cell metrics indexed by (cell, P, V, T).
cell	str	Required	Name of the target cell for metric extraction
pvt	tuple	Required	Tuple representing the Process, Voltage, and Temperature conditions
xy	tuple	From INFO.json	(load, tran) condition for interpolation
xyrange	tuple	From INFO.json	Range of (load, tran) values for surface regression
figsize	tuple	(8,6)	Plot size (width, height) for visualization

Example Code:

```
cellMetricConstraint(dtco.df,cell='cell_name_1',pvt=('FFG',0.43,25,'cbest'),xy=dtco.param[
'xy'],xyrange=dtco.param['xyrange'],figsize=(8,6))
```

6.3. batchMetricPPA

Purpose: Evaluates the Performance, Power, and Area (PPA) metrics for a batch of selected cells using extracted metric values from the input DataFrame. This function enables users to compare PPA across multiple cells and PVT corners for design optimization.

Visualization Types: Generates comparative visualizations to show metric trends across different PVT conditions for multiple cells, highlighting performance variations under various (load, tran) conditions.

Comparison: Facilitates cross-probing of cells across multiple PVT corners to evaluate how each cell's metrics (performance, power, area) respond to changing conditions. Users can assess design trade-offs efficiently through visual trends.

Data Filtering: Utilizes referenced (load, tran) index ranges to refine the surface regression analysis. The results focus on specific cross-probing scenarios, enabling targeted PPA evaluations.

Parameters:

Parameter	Type	Default	Description
dt	Pandas.DataFrame	Required	DataFrame containing cell metrics indexed by (cell, P, V, T).
cellL	list[str]	Required	List of cells for PPA evaluation
pvtL	list[tuple]	Required	List of tuples representing PVT corners
xy	tuple	From INFO.json	(load, tran) condition for cross-probing
xyrange	tuple	From INFO.json	Range of (load, tran) values for surface regression
figsize	tuple	(10,6)	Plot size (width, height) for visualization

Example Code:

```
batchMetricPPA(dtco.df,cellL=['cell_name_1','cell_name_2','cell_name_3'],pvtL=[('FFG',0.43
,25,'cbest'),('FFG',0.43,85,'cbest'),('SSG',0.45,25,'cworst')],xy=dtco.param['xy'],xyrange
=dtco.param['xyrange'])
```

6.4. batchMetricTimingBalance

Purpose: Evaluates the timing imbalance for multiple cells under specified PVT conditions and (load, tran) parameters. This function leverages Least Squares (LS) regression to interpolate timing metrics, allowing users to identify transition or delay imbalances across different operating conditions.

Visualization Types: Generates comparative visualizations highlighting timing imbalance trends under various (load, tran) conditions. Supports both delay and transition visualizations to offer insights into how different cells behave under dynamic loads.

Comparison: Allows users to assess the balance between transition and delay across multiple cells and PVT corners, helping designers make more informed decisions about timing optimizations.

Data Filtering: Utilizes the ctype parameter to select the relevant metric type: 'cell' for delay or 'tran' for transition. This ensures that the analysis focuses only on the desired performance aspect.

Parameters:

Parameter	Type	Default	Description
dt	Pandas.DataFrame	Required	DataFrame containing cell metrics indexed by (cell, P, V, T).
cellL	list[str]	Required	List of cells for timing imbalance evaluation
pvtL	list[tuple]	Required	List of tuples representing PVT corners
xy	tuple	From INFO.json	(load, tran) condition for interpolation
xyrange	tuple	From INFO.json	Range of (load, tran) values for LS regression
ctype	str	'tran'	Metric type: 'cell' for delay, 'tran' for transition
figsize	tuple	(8,6)	Plot size (width, height) for visualization

Example Code:

```
batchMetricTimingBalance(dtco.df,cellL=['cell_name_1','cell_name_2','cell_name_3'],pvtL=[(
'FFG',0.43,25,'cbest'),('FFG',0.43,85,'cbest'),('SSG',0.45,25,'cworst')],xy=dtco.param['xy
'],xyrange=dtco.param['xyrange'])
```

6.5. batchMetricConstraint

Purpose: Evaluates cell constraint metrics for multiple cells using Least Squares (LS) regression based on (clock, data) conditions. This function helps designers assess how timing constraints vary across different PVT corners and conditions, aiding in the optimization of design constraints.

Visualization Types: Generates visualizations of constraint metrics, including clock vs. data trends, to provide a comprehensive view of constraint behavior under multiple conditions.

Comparison: Allows users to compare constraint metrics across multiple cells and PVT corners, providing insights into how cells respond to clock and data variations. This comparison ensures that designs meet timing requirements under various scenarios.

Data Filtering: Filters timing arcs using regular expressions to ensure only relevant data points are included for analysis. This optimizes regression accuracy and visualization clarity.

Parameters:

Parameter	Type	Default	Description
dt	Pandas.DataFrame	Required	DataFrame containing cell metrics indexed by (cell, P, V, T).
cellL	list[str]	Required	List of cells for timing imbalance evaluation
pvtL	list[tuple]	Required	List of tuples representing PVT corners
xy	tuple	From INFO.json	(clock, data) condition for interpolation
xyrange	tuple	From INFO.json	Range of (clock, data) values for LS regression
interp	bool	True	Flag to toggle between LS regression (True) or direct metric values (False)
figsize	tuple	(10,6)	Plot size (width, height) for visualization

Example Code:

```
batchMetricConstraint(dtco.df,cellL=['cell_name_1','cell_name_2','cell_name_3'],pvtL=[('FF
G',0.43,25,'cbest'),('FFG',0.43,85,'cbest'),('SSG',0.45,25,'cworst')],xy=dtco.param['xy'],
xyrange=dtco.param['xyrange'])
```

6.6. batchDelayInsertion

Purpose: Calculates the required cell stages and associated penalties (such as area usage and setup margin) to achieve a specified target delay. This function helps designers select appropriate cells and evaluate trade-offs for achieving timing closure across PVT conditions.

Visualization Types: Generates visualizations showing the delay insertion trends and the impact on area and setup margin across different (load, tran) conditions and PVT corners.

Comparison: Allows users to compare multiple cell candidates and their impact on the target delay, evaluating both the best-case and worst-case PVT corners to ensure robust timing.

Data Filtering: Uses LS regression to interpolate delay trends based on the given (load, tran) conditions, helping users fine-tune the required number of cell stages and evaluate penalties.

Parameters:

Parameter	Type	Default	Description
dt	Pandas.DataFrame	Required	DataFrame containing cell metrics indexed by (cell, P, V, T).
cellL	list[str]	Required	List of candidate cells for delay insertion
targetD	float	Required	Tuple representing the Process, Voltage, and Temperature conditions
pvtB	tuple	Required	(load, tran) condition for interpolation
pvtW	tuple	Required	Range of (load, tran) values for surface regression
xy	tuple	From INFO.json	Regular expression to filter timing arcs
xyrange	tuple	From INFO.json	Range of (load, tran) values for LS surface regression
figsize	tuple	(10,6)	Plot size (width, height) for visualization
weight	float	0.8	Weight factor for balancing penalties (area vs. setup margin)

Example Code:

```
batchDelayInsertion(dtco.df,cellL=['cell_name_1','cell_name_2','cell_name_3'],pvtL=[('FFG'
,0.43,25,'cbest'),('FFG',0.43,85,'cbest'),('SSG',0.45,25,'cworst')],xy=dtco.param['xy'],xy
range=dtco.param['xyrange'],weight=0.8,targetD=100)
```