

# MPVIS – A Python Library for Multi-Perspective Visualization in Process Mining

Nicolás Abarca-Quiroga<sup>1</sup>, Ignacio Velásquez<sup>1,\*</sup> and Marcos Sepúlveda<sup>1</sup>

<sup>1</sup>Department of Computer Science, School of Engineering, Pontificia Universidad Católica de Chile

## Abstract

Process models are commonly used to conduct exploratory analyses during the initial stages of BPM projects. Models depicting the process behavior can be discovered through process mining. These models allow visualizing the control-flow or performance perspectives (usually through the time dimension). However, current implementations typically allow only one perspective/dimension to be displayed at the same time. This work introduces MPVIS, a Python library that allows generating process models, particularly directly-follows graphs and directed rooted trees, that visualize multiple perspectives and performance dimensions simultaneously. The library is expected to be useful for conducting exploratory analyses during the initial phases of the BPM lifecycle by enabling the identification of multidimensional insights and improvement opportunities.

## Keywords

Process mining, Process discovery, Performance analysis, Multi-perspective visualization

Metadata description	Value
Tool name	MPVIS
Current version	1.0.3
Legal code license	GNU Affero General Public License v3.0
Languages, tools and services used	Python3, PM4Py, Graphviz
Supported operating environment	Microsoft Windows, GNU/Linux, macOS
Download/Demo URL	<a href="https://bit.ly/mpvis-tutorial">https://bit.ly/mpvis-tutorial</a>
Documentation URL	<a href="https://bit.ly/mpvis-doc">https://bit.ly/mpvis-doc</a>
Source code repository	<a href="https://github.com/nicoabarca/mpvis">https://github.com/nicoabarca/mpvis</a>
Screencast video	<a href="https://bit.ly/mpvis-tutorial-video">https://bit.ly/mpvis-tutorial-video</a>

## 1. Introduction & Motivation

Process mining (PM) is a discipline that enables the analysis of processes based on event logs recording their executions [1]. One of its main tasks is process discovery, in which models depicting the observed process behavior are derived from event data [2]. These models typically allow the visualization of the control-flow perspective, revealing alternate execution paths and rework cycles, and can be extended to incorporate additional performance information such as activity frequency, duration [3], or cost [4].

*ICPM Doctoral Consortium and Demo Track 2025, October 20-24, 2025, Montevideo, Uruguay*

\*Corresponding author.

✉ [nicoabarca@uc.cl](mailto:nicoabarca@uc.cl) (N. Abarca-Quiroga); [iavelasquez@uc.cl](mailto:iavelasquez@uc.cl) (I. Velásquez); [marcos.sepulveda@uc.cl](mailto:marcos.sepulveda@uc.cl) (M. Sepúlveda)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Although multiple process performance dimensions, like time, cost, quality, and flexibility, are well established in the literature [5], most existing tools and methods allow only a single perspective or performance dimension to be displayed at a time. Some systems enable secondary metrics to be included in nodes and arcs [6] [7], but the visualization still emphasizes a primary metric, with color encoding and visual emphasis determined solely by it. This limitation impedes the ability to identify trade-offs and interactions between different perspectives.

To address this gap, we present MPVIS, a Python library for the simultaneous visualization of multiple perspectives and performance dimensions within a single process model. MPVIS implements both Directly-Follows Graphs (DFGs) [1] and Directed Rooted Trees (DRTs) [8], extending them to allow multi-perspective and multidimensional analyses. By enabling this form of integrated visualization, MPVIS supports richer exploratory analyses during the early phases of the BPM lifecycle, potentially revealing improvement opportunities that might remain hidden if perspectives are analyzed separately.

## 2. Related Work

Multi-perspective process discovery has been addressed in several studies, including clustering approaches that integrate the control-flow, data, and time perspectives [9], and dashboards that summarize multiple perspectives based on process stages [10]. In process model visualization, some works have proposed models incorporating both the control-flow and data perspectives [11], annotated with performance and frequency information [12], or enriched with context-aware operators [13].

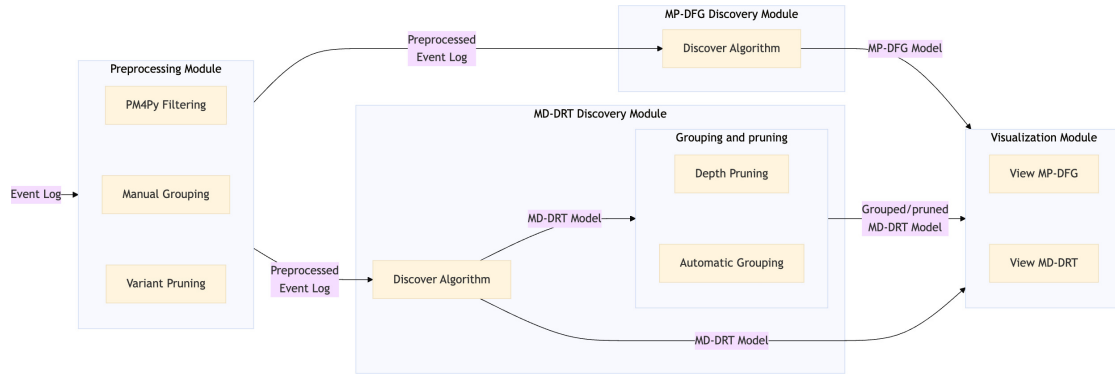
Despite these advances, existing process mining tools such as PM4Py [14], Celonis [7], Aprimore [15], and bupaR [6], either restrict the visualization to a single primary metric or limit secondary metrics to auxiliary textual annotations. Moreover, while DFGs are widely implemented, DRTs remain a recently proposed model and are not supported in most tools, with the only known implementation focusing solely on cost metrics [8].

## 3. Library Overview

MPVIS is implemented as a Python library compatible with PM4Py. Its architecture is structured around three main modules: preprocessing, model discovery, and visualization. The overall architecture is illustrated in Figure 1.

Event logs can first be preprocessed using MPVIS's own grouping, filtering and pruning functions or using PM4Py's native filters, which reduce the complexity of the resulting models by aggregating activities or reducing variants.

From these event logs, MPVIS can generate two types of visualizations. The Multi-Perspective DFG (MP-DFG) extends the traditional DFG by splitting each node into multiple sections, each encoded with a color range representing a different perspective or performance dimension, such as control-flow frequency, activity duration, and activity cost. Similarly, arcs display numerical annotations corresponding to the frequency and waiting times of arcs. Figure 2 (b) visualizes a discovered Multi-Perspective DFG, containing control-flow, time, and cost information of activities for the event log in Figure 2 (a).



**Figure 1:** Architecture of the MPVIS library.

The Multi-Dimensional DRT (MD-DRT) applies a similar strategy but focuses on representing all process variants within a single acyclic graph, with state nodes segmented into sections representing time, cost, quality (measured as the number of rework activities in cases), and flexibility (measured as the number of optional activities in cases). This approach enables the joint analysis of variant-level behavior across dimensions. MPVIS also includes functionality to aggregate non-bifurcating paths and prune the tree’s depth to manage complexity, which is particularly valuable for large or highly variable logs. Figure 2 (c) visualizes a discovered MD-DRT containing information for the time, cost, quality, and flexibility dimensions for the event log in Figure 2 (a).

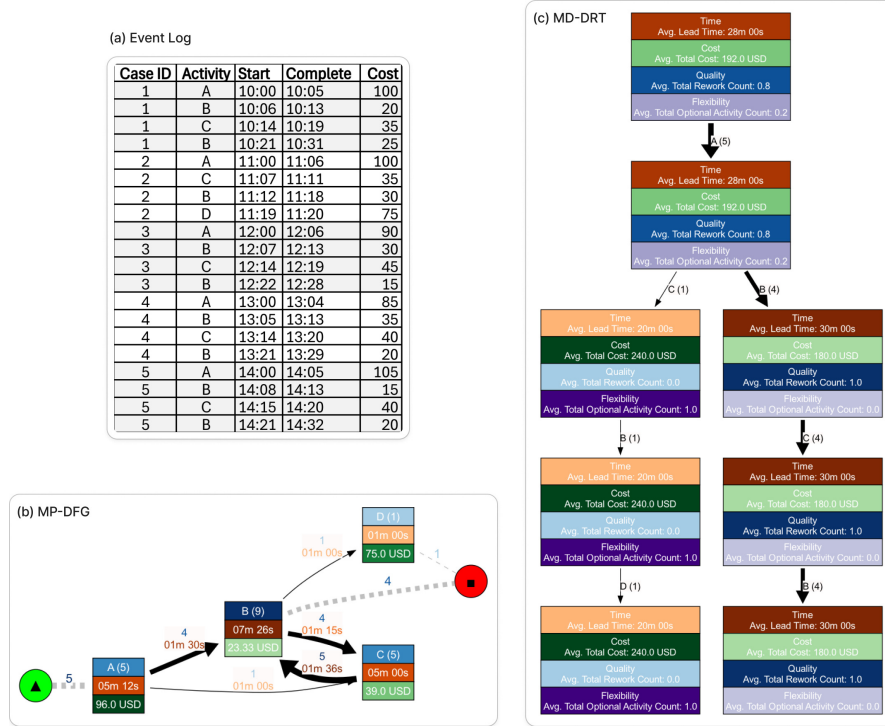
To facilitate their use, both the MP-DFG and MD-DRT discovery and visualization functions were created to the likeness of PM4Py’s functionalities, as outlined in Listing 1.

```
# Read event log using pandas utilities, then format it using MPVIS
event_log = pd.read_csv('event_log.csv', sep = ';')
log_format = {'case:concept:name': 'case_id', 'concept:name': 'activity',
              'time:timestamp': 'complete', 'start_timestamp': 'start', 'cost:total': 'cost'}
event_log = mpvis.log_formatter(event_log, log_format)
# Discover and then view MP-DFG (sa = start activities, ea = end_activities)
dfg, sa, ea = mpvis.mpdfg.discover_multi_perspective_dfg(event_log)
mpvis.mpdfg.view_multi_perspective_dfg(dfg, sa, ea)
# Discover and then view MD-DRT
drt = mpvis.mddrt.discover_multi_dimensional_drt(event_log)
mpvis.mddrt.view_multi_dimensional_drt(drt)
```

Listing 1: Minimal code for process model discovery and visualization

## 4. Features & Innovations

MPVIS’s novelty lies in its ability to visualize multiple process perspectives and performance dimensions simultaneously, while maintaining the interpretability of single perspective process mining visualizations. For the MP-DFG, this is achieved through a multi-section coloring



**Figure 2:** Visualizations of a multi-perspective DFG and a multidimensional DRT for a sample event log.

scheme for nodes and arcs. Nodes are split into three sections, each represented by a different color range: blue is used for representing the control-flow perspective (activities' frequency), red is used for representing the time dimension (activities' duration), and green is used for representing the cost dimension (activities' cost). Arcs contain blue numbers representing the control-flow perspective (frequency of directly-follows activity relations) and red numbers representing the time dimension (activities' waiting times). Distinct aggregation metrics can be configured for every perspective: absolute and relative frequencies at the activity or case level for the control-flow perspective, and mean, median, total, maximum, minimum, or standard deviation for the time and cost dimensions.

For the MD-DRT, a multi-section coloring scheme is also used for nodes. Each section of the nodes aggregates the information (total, accumulated, remaining values) of cases that flow through it and is represented by a different color range: red is used for representing the time dimension (lead time of cases), green is used for representing the cost dimension (total cost of cases), blue is used for the quality dimension (number of rework activities of cases), and purple is used for the flexibility dimension (number of optional activities of cases). An activity execution is considered as rework if it has been previously executed during a case [5]. An activity is optional if it does not occur in at least one case [5]. Arcs contain numbers indicating the information of activities flowing through them: their frequency, their service time, their cost, and whether they are considered rework and/or optional activities.

Compatibility with the PM4Py library was taken in consideration so that event logs loaded

and filtered using PM4Py can be visualized using MPVIS. This allows incorporating the functionalities of MPVIS into existing PM4Py workflows to enhance process model visualization.

Additional grouping and pruning functions have been implemented in MPVIS, which allow reducing the complexity of the resulting visualizations. Specifically, activity grouping functions that facilitate dealing with parallelism or reducing the length of a DRT's non-bifurcating paths, and pruning functions that allow limiting visualizations to a certain number of execution variants or a DRT's depth, have been implemented.

## 5. Library Maturity & Limitations

The library is currently in a stable version and no major bugs are known. Its functionalities have been tested through its application in several synthetic and publicly-available real-life event logs. The resulting process models can be found at <https://bit.ly/mpvis-examples>, and a table summarizing the considered real-life event logs and statistics related to their discovery and visualization times can be found at <https://bit.ly/mpvis-statistics>.

The maturity of the library can also be discussed in terms of its current limitations. An inherent limitation of DFGs and DRTs is that they have difficulties representing parallelism. Another inherent limitation of process models is their difficulty to visualize processes with several process variants. The grouping and pruning functions of MPVIS allow to alleviate these limitations as they reduce the resulting process model's complexity or group activities that might occur in parallel. Future work considers possibly extending the multi-perspective visualization of processes to notations that more adequately support complex behavior, such as BPMN. Another limitation is that specific process perspectives and performance dimensions were considered for both MP-DFG and MD-DRT. Future versions of the library will consider the definition of custom perspectives through user-selected aggregation metrics and color ranges.

## 6. Conclusion

This work presents a library for visualizing multiple process perspectives and performance dimensions in a single process model. Specifically, it provides a MP-DFG implementation where nodes and arcs are decorated with multiple colors to denote the control-flow perspective and the time and cost dimensions of a process, and a MD-DRT implementation where states and transitions are decorated with multiple colors to denote process performance for the time, cost, quality, and flexibility dimensions. These multidimensional visualizations allow analyzing process behavior while considering multiple perspectives in conjunction, without requiring the generation of process models for every perspective or performance dimension. This facilitates their side-to-side comparison for identifying trade-offs and other multidimensional insights.

## Acknowledgments

This work was funded by the Agencia Nacional de Investigación y Desarrollo de Chile [grant numbers ANID FONDECYT 1230697, and ANID-Subdirección de Capital Humano/Doctorado Nacional/2021-21210022].

## Declaration on Generative AI

During the preparation of this work, the authors used **ChatGPT** to: **Grammar and spelling check, Improve writing style, Paraphrase and reword**. After its use, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

- [1] W. Van der Aalst, *Process mining: data science in action*, Springer, 2016.
- [2] S. J. Leemans, E. Poppe, M. T. Wynn, Directly follows-based process mining: Exploration & a case study, in: *2019 International Conference on Process Mining*, IEEE, 2019, pp. 25–32.
- [3] R. Cao, W. Ni, Q. Zeng, F. Lu, C. Liu, H. Duan, Remaining time prediction for business processes with concurrency based on log representation, *China Communications* 18 (2021) 76–91.
- [4] M. Wynn, W. Z. Low, A. ter Hofstede, W. Nauta, A framework for cost-aware process management: cost reporting and cost prediction, *Journal of Universal Computer Science* 20 (2014) 406–430.
- [5] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, *Fundamentals of business process management*, Springer, 2013.
- [6] G. Janssenswillen, B. Depaire, M. Swennen, M. Jans, K. Vanhoof, bupar: Enabling reproducible business process analysis, *Knowledge-Based Systems* 163 (2019) 927–930.
- [7] C. Ullrich, T. Lata, J. Geyer-Klingenberg, Celonis studio-a low-code development platform for citizen developers, in: *BPM (PhD/Demos)*, 2021, pp. 102–105.
- [8] I. Velásquez, M. Sepúlveda, A tool for visualizing costs of process variants through directed rooted trees, in: *BPM (Demos/Resources Forum)*, 2023, pp. 72–76.
- [9] Y. Bertrand, J. De Weerd, E. Serral, A novel multi-perspective trace clustering technique for iot-enhanced processes: A case study in smart manufacturing, in: *International Conference on Business Process Management*, Springer, 2023, pp. 395–412.
- [10] H. H. Nguyen, *Stage-aware business process mining*, Thesis, Queensland University of Technology, 2019.
- [11] F. M. Maggi, M. Dumas, L. García-Bañuelos, M. Montali, Discovering data-aware declarative process models from event logs, in: *Business Process Management: 11th International Conference*, Beijing, China. Proceedings, Springer, 2013, pp. 81–96.
- [12] A. Berti, W. van Der Aalst, Extracting multiple viewpoint models from relational databases, in: *International Symposium on Data-Driven Process Discovery and Analysis*, Springer, 2018, pp. 24–51.
- [13] R. Shraga, A. Gal, D. Schumacher, A. Senderovich, M. Weidlich, Process discovery with context-aware process trees, *Information Systems* 106 (2022) 101533.
- [14] A. Berti, S. van Zelst, D. Schuster, Pm4py: a process mining library for python, *Software Impacts* 17 (2023) 100556.
- [15] M. La Rosa, H. A. Reijers, W. M. Van Der Aalst, R. M. Dijkman, J. Mendling, M. Dumas, L. García-Bañuelos, *Apromore: An advanced process model repository*, *Expert Systems with Applications* 38 (2011) 7029–7040.