

PERSPECT: An Interactive Tool for Performance Pattern Mining

Moritz Schlegelmilch¹, Piotr Zaniwski¹,
Volodymyr Diemieniev¹, Jasper Saathoff¹,
Tsung-Hao Huang¹, and Gyunam Park²

¹ RWTH Aachen University

{moritz.schlegelmilch, piotr.zaniwski, volodymyr.diemieniev,
jasper.saathoff}@rwth-aachen.de, tsunghao.huang@pads.rwth-aachen.de

² Eindhoven University of Technology

g.park@tue.nl

Abstract. Complex processes can be difficult to understand from event data. Traditional process models effectively visualize control flows, but lack temporal context for performance analysis. To address this, the Performance Spectrum was introduced to map the dynamics of all steps and cases over time. However, its manual interpretation is challenging due to the overlapping patterns and noise. In this tool demonstration paper, we present PERSPECT, an interactive, web-based user interface that significantly enhances the original Performance Spectrum approach. To facilitate interactive analysis, the tool automates the targeted detection of established operational behaviors (e.g., batching and FIFO), alongside advanced filtering capabilities and a side-by-side comparison function for evaluating the same segment across different process variants.

Keywords: Process Mining · Performance Spectrum · Working Patterns

1 Introduction

In continuously evolving information systems, operational processes change frequently, making it difficult to maintain a stable understanding of process behavior over time. Recent process mining research addresses these challenges by identifying change periods [8] and leveraging advances in machine learning to extract insights from event logs exhibiting multiple process variants [2,7]. Despite these advances, traditional process models discovered from event logs primarily focus on control-flow relations [1], lacking a temporal dimension. While they effectively visualize the ordering of activities and reveal deviations or bottlenecks, large real-life unstable processes often result in overly complex models that fail to capture interdependencies between cases or to convey how performance evolves over time.

To address these limitations, Denisov et al. introduced the *Performance Spectrum* [4], a visual analytics technique that adds a temporal dimension by mapping the dynamics of all steps and cases over time. By visualizing individual case

trajectories between pairs of activities, the Performance Spectrum enables analysts to observe detected temporal patterns that are often hidden in traditional process models. This includes identifying behaviors such as batching, synchronization, or shifts in execution speed that may correspond to changing operating conditions present in the event log.

In this tool demonstration paper, we present PERoSPECT³, a modernized, web-based version of the original *Performance Spectrum* approach, featuring enhanced functionality. While its conceptual strengths are preserved, our software enhances exploratory capabilities. First, it can detect patterns such as batching and FIFO based on user-provided parameters, enabling automated, targeted pattern identification. Furthermore, it offers a variety of filters and statistics that allow users to conduct in-depth, isolated analyses. In addition, a comparison function enables users to compare the Performance Spectrum of the same segment across different selected variants. Last but not least, an export function enables analysis of the filtered log outside of the tool.

The remainder of the paper is structured as follows. Section 2 provides background on the Performance Spectrum, while Section 3 outlines related tools to motivate our contribution. Section 4 presents PERoSPECT, covering the core workflow for interactive pattern mining and comparison. Section 5 details the system architecture and implementation. Section 6 discusses tool maturity and provides a practical use case. Finally, Section 7 concludes the paper and outlines future work.

2 Background

Process mining techniques analyze *event logs*, which record the execution of activities for different process instances (cases). Each event contains at least a case identifier, an activity, and a timestamp. To visualize the temporal dynamics hidden within this raw data, the *Performance Spectrum* visualizes how cases move between two successive activities over time [4] (as exemplified in Fig. 2). This visualization reveals how quickly cases progress between activities and exposes characteristic operational patterns. For a pair of activities (a, b) , each occurrence where a is directly followed by b in a case is plotted as a line connecting the timestamp of a and the timestamp of b . Time is shown on the x-axis, while the two activities are placed on the y-axis.

One of the most prominent behaviors visible in the Performance Spectrum is *batch processing*, where multiple cases are handled within a short time period [10]. Prior work distinguishes three types of batching behavior [4,10]:

- **Simultaneous**: multiple cases start and finish the activity at the same time.
- **Sequential (FIFO)**: cases are processed one after another in arrival order.
- **Concurrent**: cases overlap in execution but are not strictly sequential.

³ A public repository is available at <https://github.com/moritzkschlegelmilch/PerSpect>.

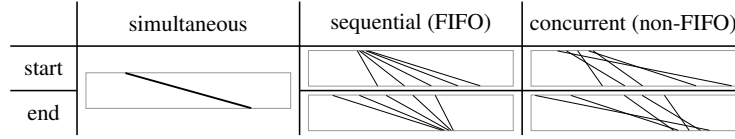


Fig. 1: Different batching patterns in the Performance Spectrum.

These batching behaviors appear as characteristic structures in the Performance Spectrum. Simultaneous batches share identical timestamps, sequential batches form ordered non-crossing lines, while concurrent batches allow overlapping and crossing lines. The aforementioned patterns are visualized in Fig. 1.

3 Related Work

To motivate our contribution, Fig. 2 illustrates an analysis performed using existing process mining tools [6,11]. The left side illustrates a directly-follows graph discovered from a real-life event log [9], while the right side presents six Performance Spectrum segments, each representing a pair of successive process steps. Notably, the segment (*Add Penalty: Send for Credit Collection*) exposes batching behavior that remains hidden when looking solely at the process model.

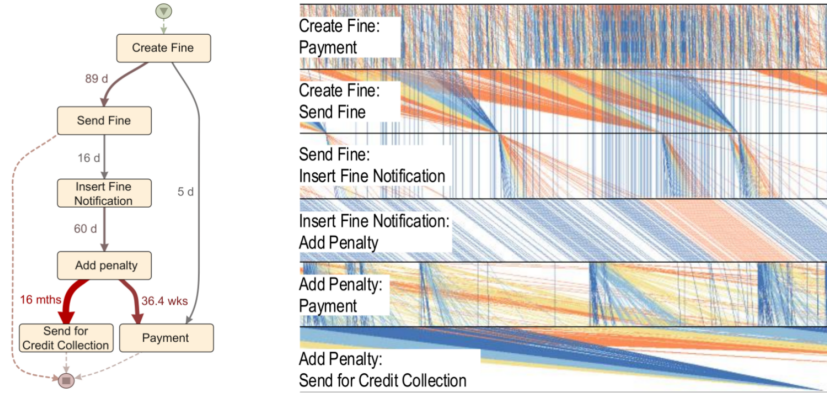


Fig. 2: Example of a process model with the corresponding performance spectrum [4].

Despite its analytical benefits, interpreting the Performance Spectrum for complex systems remains challenging due to visual clutter from overlapping patterns, noise, and irregular behavior. For example, within the (*Create Fine: Send Fine*) segment in Fig. 2, most cases exhibit a batching pattern, yet others execute the subsequent activity immediately. This entanglement makes it difficult to visually compare behaviors.

The original *Performance Spectrum Miner* [4], implemented as a ProM plugin, demonstrated that visualizing execution flows over time uncovers phenomena like workload fluctuations, delays, batching, and synchronization effects. These insights are typically lost in traditional process models or aggregated performance metrics. This early tool provided core functional capabilities, including segment filtering, case ID selection, throughput specification, and duration classifiers. However, it treats the spectrum primarily as an end-stage visualization rather than the foundation for a structured analysis workflow, lacking the ability to export filtered event logs. Consequently, analysts are forced to manually disentangle overlapping timelines and visually interpret patterns—a task requiring significant domain expertise.

Recently, Celonis introduced a modernized iteration of the Performance Spectrum [3]. While it offers a user-friendly interface and highly configurable segment organization, extracting insights still relies on manual pattern inspection. PERSPECT bridges this gap by positioning the spectrum as the foundation of a pattern-oriented workflow. The novelty of the tool lies in seamlessly integrating automated pattern extraction directly into the visual interface, enabling users to dynamically filter, isolate, and compare specific execution behaviors on the fly. This targeted approach enables the investigation of temporal behaviors that would otherwise remain obscured in aggregated data.

4 Functionality

PERSPECT empowers analysts to interactively filter and explore temporal execution patterns, disentangling complex, overlapping behaviors such as batching and near-immediate transitions. The following paragraphs detail the core capabilities⁴ of the tool.

Import & Export. To start, users are asked to upload event logs in the **.xes** (eXtensible Event Stream) format [12]. When users upload a file, the system automatically infers standard column mappings (e.g., **concept:name** for activities), though users can optionally map the essential **Case ID**, **Activity**, and **Timestamp** columns manually [1]. Once the data is uploaded, users can analyze it using the following functions. A subset of the event log, filtered for the cases in the current spectrum, can also be exported as a **.xes** file for further analysis.

Segment Picker. The Segment Picker allows users to filter the spectrum for a specific segment, which is defined as a pair of directly following activities (e.g., activity A and activity B) [4]. The starting point of a line on the spectrum corresponds to the start of the first activity (A), and the end of the line corresponds to the start of the second activity (B). The segments can be stacked to reflect the progress of the underlying process.

⁴ A video introducing the main functionalities is publicly available online <https://youtu.be/GhrcORMiQt4>.

Filtering. The tool provides various filtering options, which are divided into two types: *Global Filters* and *Segment-specific Filters*. Global Filters affect the entire spectrum and all cases. These include:

- Variant Filter: The variants view allows the users to focus on a specific variant of the process for further analysis. Once the variants are selected, the performance spectrum will only show cases with the matching process flows.
- Date-Range Filter: Users can select a specific time period to focus on within the event log. This can be done by using a calendar popover or by dragging a time window directly onto the spectrum.
- Quartile Filter: This filter visualizes cases by duration quartiles, using color codes from light blue (fastest) to red (slowest). This filter is also part of the original tool developed in ProM [4].

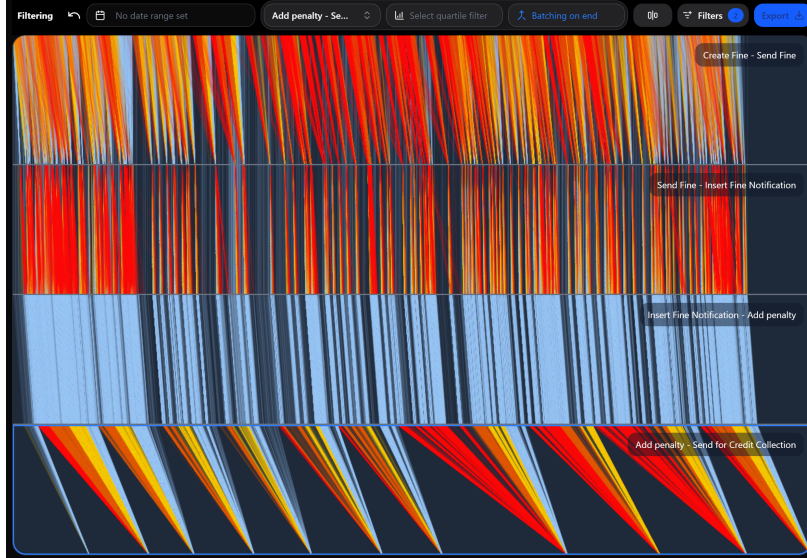


Fig. 3: An example interface of filtering by variants.

Segment-specific Filters apply only to the selected spectrum. These include:

- Batch Filters: One of the highlights of the tool is the ability to identify and extract organizational patterns from the performance spectrum, even if they are overlapping with other cases. For example, batches are sets of cases that are grouped based on their start or end time using a clustering technique. In principle, any clustering algorithm that is robust to noise can be used. In the tool, we use the DBSCAN-clustering algorithm [5] as an example to realize our pattern extraction function. Users can filter for batches that start, end, or

both start and end around the same time [10]. Furthermore, users can adjust the clustering parameters, **Epsilon** (how close the cases must be in time to belong to the same batch) and **Min-Samples** (the minimum number of cases required for a batch to be detected). Fig. 4 shows an example of a segment with overlapping patterns and how different patterns are extracted and visualized separately. Notably, the tool allows setting **Epsilon** to 0 minutes, which enables the automated discovery of *simultaneous* batches. Lastly, the tool distinguishes between FIFO (first in, first out) and non-FIFO batching patterns.

- **Interactive Filter:** The interactive filter allows users to interactively select cases of interest within a segment. While the original ProM plugin merely highlighted such selections, our implementation dynamically updates all other spectra to reflect the chosen cases.

The system automatically tracks every filtering action performed by the user. This enables the user to recover a prior state of the spectrum.

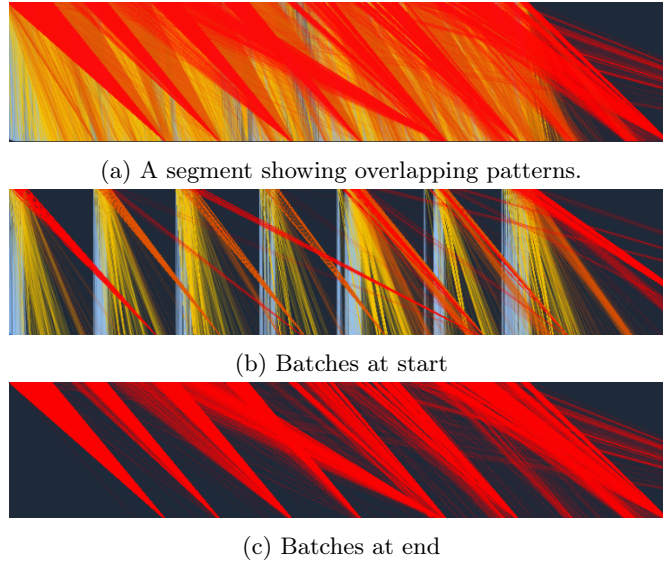


Fig. 4: An example showing different patterns of the same segment.

Comparative Analysis. The tool enables direct, side-by-side comparison of filtered spectra by allowing users to create and assign meaningful names to multiple configurations. Each configuration saves the specific state of a spectrum. Users can seamlessly switch between these configurations to compare different filtered views, or easily remove them when no longer needed. To ensure readability, the viewport is designed to display a maximum of two configurations side-by-side at any given time. Fig. 5 shows an example view of the comparative analysis.

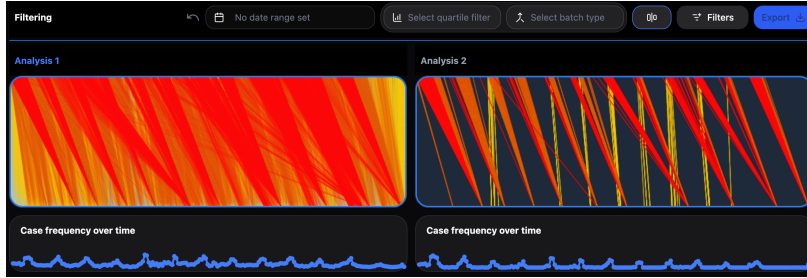


Fig. 5: An example interface of the comparative analysis.

Process & Pattern Statistics. To help analyze detected patterns, the tool provides several quantitative metrics and diagrams.

- Frequency Diagrams: Located below the Performance Spectrum, these diagrams show how frequently cases start and end over time, displaying the number of cases per time bin. Users can hover over points on the diagram for more details.
- Basic Statistics: This section provides metrics for the current selection, including the number of cases, distinct activities, distinct variants, and duration mean.
- Batch Statistics: When batch filtering is active, this section displays information such as the total number of batches, average batch size, average batch duration, average batch interval, and the percentage of cases in the entire event log that are part of a batch.

5 Architecture

The system follows a client-server architecture consisting of a backend for event log management and spectrum computation, and a web-based frontend for interactive analysis.

Backend Services and Spectrum Processing. The backend exposes a REST API implemented with FastAPI. Event logs are uploaded to the server, while metadata (name, storage path, and attribute mappings) is stored in a lightweight SQLite database managed via Alembic. For performance, loaded event logs are cached in memory using a Least Recently Used (LRU) strategy, and concurrent loading is synchronized to avoid redundant operations. Spectrum generation follows a query-based pipeline that separates global filters from segment-specific filters. Global filters (e.g., case selection, time range, segment or variant selection) determine the base spectra, after which segment-specific filters are applied. Filters are executed in a controlled order to ensure consistent results, and case sets are synchronized across spectra by intersecting case identifiers.

Frontend. The frontend is implemented as a Vue-based single-page application and introduces *configurations* as independent analysis states that can be displayed side-by-side. Each configuration maintains its own filter history and submits a unified filter payload to the backend to retrieve spectra or export the filtered event log.

6 Maturity

PERSPECT is a fully documented open-source research prototype. Designed for integration into existing process mining pipelines, it supports the de facto standard `.xes` format [12] for event log import and export. The tool’s maturity lies in transforming the Performance Spectrum from a purely exploratory visualization into a systematic, pattern-oriented analytical workflow. By automating the detection of temporal behaviors—such as simultaneous, FIFO and non-FIFO batching—and enabling isolated, side-by-side comparisons, PERSPECT supports both academic research and practical application in analyzing event data.

Use Case Example. To demonstrate these capabilities in practice, the Road Traffic Fine Management process is used [9]. An analyst investigating delays might focus on the (*Add Penalty: Send for Credit Collection*) segment shown in Fig. 3. Using the Batch Filter, PERSPECT reveals that over 97% of cases follow a sequential batching pattern where only 11 batches occur with an average interval of 410 days. This indicates that cases are held in a long-term backlog and cleared in massive, infrequent bursts. PERSPECT provides the visual and numerical evidence to reveal these patterns, providing a level of transparency that control-flow models typically fail to capture.

7 Conclusion

We developed PERSPECT to overcome the interpretation challenges caused by visual clutter, overlapping temporal patterns, and noise in complex event logs. The tool delivers a modern, web-based instantiation of the original Performance Spectrum concept. By introducing the automated algorithmic detection of organizational patterns, advanced filtering for isolated analysis, and comparative viewport configurations, PERSPECT empowers analysts to uncover structural performance behaviors that typically remain hidden within aggregated metrics.

Future work will focus on expanding the filter selection to include attribute filters, capturing a broader range of execution behaviors. Additionally, we aim to integrate more advanced visual summaries to deepen insights into temporal dynamics. Finally, we plan to explore side-by-side integrations with control-flow visualizations, such as BPMN models or directly-follows graphs. By visually linking a selected spectrum segment directly to its corresponding activities within an overall process model, users will gain immediate structural context for their performance analysis.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016). <https://doi.org/10.1007/978-3-662-49851-4>
2. Cai, R., Zheng, C., Wang, J., Li, D., Wang, C., Li, B.: Reinforcement learning-based streaming process discovery under concept drift. In: Guizzardi, G., Santoro, F.M., Mouratidis, H., Soffer, P. (eds.) Advanced Information Systems Engineering - 36th International Conference, CAiSE 2024, Limassol, Cyprus, June 3-7, 2024, Proceedings. pp. 55–70. Lecture Notes in Computer Science, Springer (2024). https://doi.org/10.1007/978-3-031-61057-8_4
3. Celonis SE: Performance Spectrum. <https://docs.celonis.com/en/performance-spectrum.html> (2026), accessed: 2026-04-09
4. Denisov, V., Fahland, D., van der Aalst, W.M.P.: Unbiased, fine-grained description of processes performance from event data. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) Business Process Management - 16th International Conference, BPM 2018, Sydney, NSW, Australia, September 9-14, 2018, Proceedings. pp. 139–157. Lecture Notes in Computer Science, Springer (2018). https://doi.org/10.1007/978-3-319-98648-7_9
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. pp. 226–231. KDD’96, AAAI Press (1996), <https://cdn.aaai.org/KDD/1996/KDD96-037.pdf>
6. Günther, C.W., Rozinat, A.: Disco: Discover your processes. In: Lohmann, N., Moser, S. (eds.) Proceedings of the Demonstration Track of the 10th International Conference on Business Process Management (BPM 2012), Tallinn, Estonia, September 4, 2012. pp. 40–44. CEUR Workshop Proceedings, CEUR-WS.org (2012), <https://ceur-ws.org/Vol-940/paper8.pdf>
7. Kraus, A., van der Aa, H.: Machine learning-based detection of concept drift in business processes. *Process Sci.* **2**(1) (2025). <https://doi.org/10.1007/S44311-025-00012-W>
8. Kraus, A., Elyasi, K.A., van der Aa, H.: On the use of steady-state detection for process mining: Achieving more accurate insights. In: Krogstie, J., Rinderle-Ma, S., Kappel, G., Proper, H.A. (eds.) Advanced Information Systems Engineering - 37th International Conference, CAiSE 2025, Vienna, Austria, June 16-20, 2025, Proceedings, Part I. pp. 204–220. Lecture Notes in Computer Science, Springer (2025). https://doi.org/10.1007/978-3-031-94569-4_12
9. de Leoni, M.M., Mannhardt, F.: Road traffic fine management process (2015). <https://doi.org/10.4121/UUID:270FD440-1057-4FB9-89A9-B699B47990F5>
10. Martin, N., Swennen, M., Depaire, B., Jans, M., Caris, A., Vanhoof, K.: Retrieving batch organisation of work insights from event logs. *Decision Support Systems* **100**, 119–128 (2017). <https://doi.org/10.1016/j.dss.2017.02.012>, Smart Business Process Management
11. Verbeek, E., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: ProM 6: The process mining toolkit. In: Rosa, M.L. (ed.) Proceedings of the Business Process Management 2010 Demonstration Track, Hoboken, NJ, USA, September 14-16, 2010. CEUR Workshop Proceedings, CEUR-WS.org (2010), <https://ceur-ws.org/Vol-615/paper13.pdf>
12. Verbeek, E., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES Tools. In: Soffer, P., Proper, E. (eds.) Proceedings of the CAiSE Forum 2010, Hammamet, Tunisia, June 9-11, 2010. CEUR Workshop Proceedings, CEUR-WS.org (2010), <https://ceur-ws.org/Vol-592/PaperDemo07.pdf>