# Comparative Study of Trace Clustering and Process Cube Sequences in Process Mining

BYEONGHUN PARK, University of Twente, Enschede, The Netherlands

One of the key challenges that process mining often encounters is "spaghetti-like" process models generated in process discovery due to complex variants of the event data. To address this, segmentation techniques such as trace clustering and Process Cube analysis have been proposed. This study presents a comparative benchmark of these two approaches, both individually and in combination, applied to the 4TU Sepsis Cases event log. By segmenting the data using trace clustering, Process Cube, Process Cube followed by trace clustering (Process Cube → trace clustering), and trace clustering followed by Process Cube (trace clustering → Process Cube), the resulting process models are evaluated through complexity metrics such as Cyclomatic Number, Coefficient of Network Connectivity, and arc density. The results show that the Process Cube → trace clustering combination produces the most simplified and interpretable process models. The result also suggests that dimension-based segmentation provides a more effective segmentation for further clustering. This work focuses on the segmentation order and provides a methodological basis for applying appropriate techniques to improve model quality in process discovery.

Additional Key Words and Phrases: Process mining, Process cube, Trace Clustering

## 1 INTRODUCTION

### 1.1 Background and Context

Process mining uses event log data from information systems to discover and improve real-world operational processes. The type of process mining to produce a process model is called process discovery. Process discovery is based on event logs, which are formed with a collection of traces. Event logs contain attributes such as case id, activity, time stamp and other properties. Although it is not mandatory for all event logs to have identical attributes, events with a shared activity tend to share similar sets of attributes [1]. Such variability is common in real-world event logs, such as those from clinical environments, making them well-suited for analysis techniques that account for contextual diversity, including process cube analysis and trace clustering. However, one limitation of process mining is the generation of "spaghetti-like" process models when dealing with unstructured processes [8]. This challenge is notable because process discovery often results in models that are overly complex and unreadable, particularly for practical data with high variability [19].

To address this limitation, there are some approaches to segment the data for simplicity of the process models. One way to achieve this is to perform trace clustering, which segments the data based on similar activity traces. This allows each cluster composed of similar traces to be distinct with its own features [9, 19]. Trace clustering is known to decrease process model complexity by forming distinct clusters. There are numerous studies on efficient clustering with different approaches such as algorithmic approaches and those focusing on computational complexity. In practice, clustering algorithms like k-means have shown effectiveness in simplifying complex process models derived from highly variable event logs [14].

Another approach is the Process Cube, which allows different perspectives on the data by assigning multiple dimensions [2]. There are three basic concepts of the Process Cube: Event Base, Process Cube Structure, and Process Cube View. Event Base refers to a collection of event logs with mixed traces, but related to each other. Process Cube Structure defines all the possible dimensions within the cube such as time, weather, or location. Attributes can form different dimensions depending on the domain context. Importantly, each dimension in the Process Cube can also have a hierarchical structure, which is defined as a directed acyclic graph (DAG) over the attributes. These hierarchical relationships enable flexible change in granularity when analyzing event data, which makes it possible to conduct context-specific segmentation. The Process Cube View visualizes the portions of the event data by selected dimensions. Common operations on the Process Cube include slicing for filtering the event data by chosen attributes (e.g., city : Enschede), dicing for reducing the cube into a smaller scope (e.g., month : March, April), rolling up for a generalized view of the data (e.g., city → country → continent), drilling down for breaking down certain dimensions into detailed attributes (e.g., year → month → day) [2, 7].

Machine learning-based clustering and the Process Cube have clear differences in how they segment data: clustering groups traces based on similarity, while the Process Cube provides a structure of segmentation based on multidimensional attributes. A previously proposed generic framework for trace clustering of event logs, presented in [19], serves as the structural basis for segmentation in this work. Although this framework is originally designed for trace clustering, its initial stages such as feature generation and feature transformation are in line with the creation of dimensions and attributes in the Process Cube. However, the specific features generated for clustering differ in terms of their underlying characteristics and intended use from the dimensions of the process cube.

Each of these segmentation techniques has a distinct mechanism and focus, and their application to event logs may result in significantly different process models in terms of complexity and interpretability.

*TScIT 43, July 4, 2025, Enschede, The Netherlands*

### 1.2 Problem Statement

There have been many studies regarding the application of clustering in order to reduce complexity of the process model [10, 18, 3] and improve clustering efficiency [5, 19, 16, 17], while the Process Cube has also been introduced as an approach to solve complex process variants [2]. However, comparative studies benchmarking and applying these two approaches for efficient segmentation remain underexplored. Therefore, the effect of the order and type of segmentation on model quality is uncertain.

### 1.3 Objective and Goals

The goal of this paper is to propose a benchmark for comparison of the Process Cube (PC) with conventional machine learning-based trace clustering (TC) techniques in terms of their effect on reducing the complexity of the process model. In this context, benchmarking refers to a comparative evaluation of process models generated from four segmentation configurations (PC, TC, PC→TC, TC→PC), using structural complexity metrics introduced in Section 3.8. The evaluation considers individual and sequential combinations of two segmentation methods.

To achieve the goal, the study investigates four process discovery experiments using the following configurations: TC, PC, TC followed by PC (TC→PC), and PC followed by TC (PC→TC) to generate process models. The quality of discovered models will be evaluated using criteria derived from a study that reviews techniques for handling complex process models [15].

### 1.4 Research Questions

In order to achieve these goals, one main research question and three sub-questions are formulated:

- **RQ1:** How do Process Cube-based and clustering-based segmentation approaches impact the quality of process models, in terms of structural complexity and interpretability, as evaluated through process discovery?
- **RQ1.1:** How can Process Cube and Trace Clustering be applied individually and in combination to segment event logs, in order to compare their impact on process model quality?
- **RQ1.2:** What evaluation metrics can be used to benchmark process models generated through different segmentation approaches?
- **RQ1.3:** What are the differences in process model quality resulting from different segmentation applications?

RQ1.1 is already answered in the introduction of this paper through explanations of the underlying concepts of both approaches.

The remainder of this paper is structured as follows. Section 2 provides related works on trace clustering and Process Cube usage in process mining. Section 3 describes the methodology of how benchmark had proceeded. Section 4 presents the experiments and results of applying the segmentation strategies. Section 5 discusses the findings and their implications including the limitations of the study. Finally, Section 6 concludes the paper, while suggesting directions for future work.

## 2 RELATED WORKS

The paper uses two main approaches to segment event logs for process discovery : machine learning-based trace clustering and Process Cube. While both approaches aim to reduce complexity and improve readability of discovered process models, they have differences in methodology and application.

Trace clustering is a widely studied method for grouping event logs based on behavioral similarity between traces. Early approaches introduced distance metrics and clustering algorithms such as k-means and hierarchical clustering to group similar traces [9, 14, 6]. Clustered traces based on behavioral patterns reduce complexity of the model. More works have focused on improving clustering accuracy and efficiency through context-aware methods [5], co-training frameworks [4]. A generic framework for trace clustering has been proposed to demonstrate the steps of the clustering process, including data understanding, preprocessing, and modelling [19]. Despite these advances, clustering approaches often require careful feature engineering, which can affect the resulting model qualities.

On the other hand, the Process Cube provides a multidimensional structure for analyzing event logs by slicing, dicing, rolling up, and drilling down with attribute-based dimensions, which are inspired by OLAP (Online Analytical Processing) cubes' operations [2]. The Process Cube enables segment process behavior from different perspectives with different dimensions, such as time, location, or resource usage. This approach is appropriate to domains with rich contextual data such as healthcare that allows numerous different types of dimensions to be generated [7]. The formal structure of the Process Cube includes the hierarchies over attributes, facilitating flexible granularity setting through operations to either abstract or specify the generated dimensions. Although there are a few researches dealing with practical projects in the industries, existing works chiefly focus on the theoretical and implementation aspects, with less emphasis on empirical evaluation or comparison with other segmentation techniques [12, 13].

While both approaches have been used for enhancing process mining, there is not much work mainly concentrated on their combination and comparison using consistent criteria. Previous studies have typically evaluated each method solely, focusing on algorithmic improvements or implementation efficiency. This lack of comparative analysis opens the door to questions about their relative strengths, limitations, and suitable use cases.

## 3 ANALYTICAL METHODOLOGY

The following subsections describe the preprocessing steps, the trace clustering model, the Process Cube, and the combinations of them described in detail. Most of the segmentation procedures follow a previously proposed generic framework for trace clustering [19]. Since the Process Cube also functions as a segmentation method, the same framework was applied to it as well. When the combinations of two approaches are applied, each can serve as a previous step for the other, depending on the direction of the application.

### 3.1 Event Log Description

This study used the 4TU Sepsis Cases Event Log dataset, which contains an event log that captures the patient treatment data in the hospital. The dataset consists of clinical indicators (e.g. InfectionSuspected, SIRSCritTachypnea, Hypoxie, etc), minimum requirements or control-flow perspectives (e.g. case identifiers, activities, etc.). It contains 1,050 unique case IDs, and 15,214 event data. Some attributes have fewer than 15214 values, likely because they are associated with activities that occur later in the process. Before those activities are executed, the corresponding attribute values remain empty. The dataset was originally provided in XES format, but to enable easier interaction with pandas, it was converted into CSV format using Python's PM4Py library.

### 3.2 Feature Generation

*3.2.1 Trace Clustering.* For preprocessing, the event log was sorted by timestamp and grouped by case identifier to ensure the correct order of activities within each case. A new column called 'activity_mapped' was added to the dataset with all the activities mapped to single-letter alphabetical symbols. This mapping simplifies sequence representation and makes it more efficient in storage and processing. Although this preprocessing does not reduce the actual number of n-gram combinations, it makes the sequences more compact and easier to handle during n-gram generation.

*3.2.2 Process Cube.* This study mainly explores the correlation between the two approaches and the potential applicability of their combination in future research. For this reason, the dimensions were limited to the simplest and most interpretable ones, specifically age, time, and activity. Each dimension has its corresponding attributes. For the time dimension, year, month, and day attributes were extracted from the timestamps of the event log. From the original Age column, two additional labels were extracted: age_group with labels for age ranges, and age_category with grouped sets of labels into categories such as Child, Teen, Young Adult, Adult, Middle-aged, and Senior. The original Age column remained as is. Based on the contextual meaning of each column, attributes of activity and activity_group were created to group similar activities.

### 3.3 Feature Transformation

*3.3.1 Trace Clustering.* Derived from the 'activity_mapped' column, sequences of activities were generated for each trace, and n-grams were then extracted and vectorized using the CountVectorizer library, as described in [19].

*3.3.2 Process Cube.* With the attributes and dimensions extracted during feature generation, hierarchy of the attributes within each dimension is set. The time dimension follows the hierarchical structure: day → month → year. For age dimension, Age → age_group → age_category is the hierarchy of the attributes. The activity dimension contains a hierarchy of activity → activity_group. These hierarchies allowed the construction of a process cube structure (PCS), since all the dimensions and attributes for the process cube view had been formed.

### 3.4 Feature Selection

*3.4.1 Trace Clustering.* Two approaches for feature selection were applied: the box plot method and the 1.5 Interquartile range (IQR) method. A total of 115 unique 2-grams were extracted from all traces and used as input for feature selection. Based on the box plot, data points with a frequency above 500 could be identified as outliers, as the distribution became sparse beyond that point. These outliers were removed to maintain consistency in the dataset. Therefore, 108 2-grams remained using the box plot method, while the 1.5 IQR method had 97. TruncatedSVD library was used to reduce the dimension of the vector with the n_components parameter set to 50.

*3.4.2 Process Cube.* After creating all necessary dimensions and hierarchies, a Process Cube View (PCV) was developed, along with a set of functions applicable to it, such as adjusting granularity and selecting specific values. The PCV takes the granularity level for each dimension and specific value for each attribute as input; however, attribute values may also be empty.

Four operation functions were implemented for the PCV: slicing, dicing, roll-up, and drill-down. All functions return a new PCV after application. **Slicing** requires a PCV, a target dimension, and a specific value. It fixes the target dimension to that value and removes it from the new PCV. **Dicing** takes a PCV and a set of attribute values. It allows setting multiple values at the same time, while keeping all the dimensions. **Roll-up** and **drill-down** enable the change of granularity levels within a single dimension. Both operations require a PCV and the target dimension. Roll-up moves the granularity level to a higher hierarchy, while drill-down changes the granularity level to a lower hierarchy. When roll-up or drill-down can not be performed, a PCV with no changes is returned. Change of granularity is limited to one level per operation. For example, moving from the attribute "Age" to "age_category" requires two roll-up operations.

### 3.5 Event Log Splitting Approach

The overall structure of the event log splitting approach is illustrated in Figure 1.

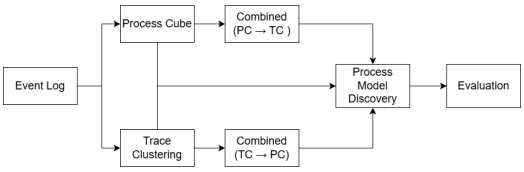


Fig. 1. Overall experimental structure of Process Cube (PC) and Trace Clustering (TC).

For the combination of PC → TC, materialized event data can be regarded as a stage of event log splitting approach, applied before clustering. Each cell represents a sub log extracted from the original event log. The sub log with the largest numbers of events was selected to reduce its complexity by clustering.

For the combination of TC → PC, the clusters generated from trace clustering can be regarded as a stage of event log splitting approach,

applied before the Process Cube operations. Each cluster represents a sub log, which makes it possible to discard cells with too few events from the Process Cube and enables operations to be performed on highly concentrated cells.

### 3.6 Trace Clustering Modelling

To explore behavioral patterns in patient treatment processes, trace clustering was performed on the event log data. The K-Means clustering algorithm was adopted due to its effectiveness in handling highly variable event logs according to [14]. Given that the dataset includes both minimal requirements and rich event attributes (32 columns in total), the use of K-Means was considered suitable for simplifying the resulting process models. Although various values for the number of clusters were considered during experimentation, the final value was fixed to k=2, as will be discussed in the Experiments and Results section.

### 3.7 Process Cube Operation

After defining the final structure of the PCV, it is materialized by inserting event data into each cell. The result of this materialization is a dictionary that maps each cell to its corresponding event data. The number of events per cell can also be checked and additional operations can be conducted.

### 3.8 Evaluation

Process discovery is performed using the inductive miner of the PM4Py library to obtain process models for evaluation. As stated in [11], inductive miner is a reliable and widely used discovery algorithm with sound and precise process models. Two functions for process discovery were defined: one for models generated after trace clustering and the other for models generated from process cube segmentation.

Evaluation of the process models has been conducted with evaluation metrics introduced in [15]. The number of nodes and arcs of the process models were used because of their simple and computational efficiency, as well as their wide adoption as metrics to indicate the visual complexity. A high number of them suggests high complexity of the model. The Coefficient of Network Connectivity (CNC) is defined as the number of arcs divided by the number of nodes in the process graph. It shows the overall density of the model. A higher CNC increases the possibility of cycles between nodes, which are more challenging to interpret than sequential paths. Place Transition Connection Degree (P/T-CD) also measures density, but focuses on the number of arcs between transitions (activity changes, shown as rectangles) and places (states, shown as circles). A high P/T-CD value implies increased spaghetti-ness of the model and reduced readability. Density of the arcs, defined as actual number of arcs/maximum possible number of arcs, is included as a normalized measure that can have a maximum value of 1. A higher arc density could imply a result of a complex model.

The Cyclomatic Number (CN) shows the number of linearly independent paths in the process model, which means a high CN means repetition or overlaps of paths exists in the model leading to increase in complexity. Average Connected Degree (ACD) computes the average number of arcs in/out from the connectors. Since it indicates the local density of arcs around connectors (transitions and places), it similarly helps quantify the complexity of connections.

The Control Flow Complexity (CFC) is the aggregate of out-degree splits (e.g. OR, XOR, AND). It is an indicator that shows how many parallel or conditional branches exist. An increase in CFC means an increase in model complexity. Finally, the number of process variants provides an idea on the diversity and variability of the behavior in process discovery. It is useful when demonstrating before and after segmentation.

Fitness, Precision, and F-Score are well-known evaluation metrics widely used in process model evaluations. However, due to their high computational cost and long execution time, they were excluded from this study.

These metrics are implemented in a self-developed evaluation function. The function takes the following inputs: (1) a dictionary of discovered models for each cluster or cell, (2) a dataframe with clustered traces, which is only used for trace-clustered models, (3) a preprocessed event log dataframe, (4) the column names for activities, (5) the column name for traces, and (6) the column name for timestamps. This structure allows modular evaluation of process complexity for different segmentation outcomes and dataset formats. The implementation of the segmentation procedures and evaluation metrics is publicly available at: https://github.com/BPark01/Data-Segmentation-Benchmark-in-Process-Mining.git.

## 4 EXPERIMENTS AND RESULTS

This section reports the results of the experiments conducted using four segmentation strategies on the Sepsis Cases event log. The complexity of the discovered process models is evaluated with the metrics, such as CNC, CN, and Arc Density, as defined in the methodology section. The following subsections present the experimental setup and the corresponding results.

### 4.1 Experimental Setup

The experiment was conducted using Python in a Jupyter Notebook environment. Multiple n-gram configurations have been experimented using CountVectorizer and TruncatedSVD. The explained_variance_ratio_.sum was used to assess how much information was retained after truncating columns of the vector. TruncatedSVD is applied to features filtered by the box plot method filtered by the box plot method, which included only data points with frequencies below 500, the explained variance was 0.9760. In contrast, the IQR method resulted in an explained variance of 0.9577, which is approximately 0.02 less than the box plot method. This means the box plot method preserves more of the original information than the IQR method. The number of clusters was set to k=2, as both the elbow method and silhouette score analysis showed that the optimal number of clusters was two, as illustrated in Figure 2. The silhouette score reached its maximum at k=2, and the elbow curve had the biggest change at the same point, which supported this decision. On one hand, when k was set to 2, K-Means clustering on features selected by the IQR method contained 872 and 178 traces in each

Table 1. Model complexity metrics for Trace Clustering (TC) and PC→TC

| Segmentation | # Arcs | # Nodes | CNC | P/T-CD | CN | ACD | Density | CFC | # Variants | # Events |
|---|---|---|---|---|---|---|---|---|---|---|
| TC (Cluster 0) | 118 | 90 | 1.311 | 2.655 | 29 | 2.622 | 0.014732 | 10 | 677 | 11239 |
| TC (Cluster 1) | 94 | 70 | 1.342857 | 2.721257 | 25 | 2.685714 | 0.019462 | 10 | 169 | 3975 |
| PC→TC (Cluster A) | 38 | 31 | 1.225806 | 2.47479 | 8 | 2.451613 | 0.040860 | 3 | 273 | 4318 |
| PC→TC (Cluster B) | 20 | 16 | 1.25 | 2.5 | 5 | 2.5 | 0.083 | 3 | 3 | 444 |

Table 2. Model complexity metrics for the Process Cube (PC) and TC→PC

| Segmentation | # Arcs | # Nodes | CNC | P/T-CD | CN | ACD | Density | CFC | # Variants | # Events |
|---|---|---|---|---|---|---|---|---|---|---|
| PC (2014, 'Senior', 'Test') | 38 | 31 | 1.222581 | 2.47479 | 8 | 2.451613 | 0.04086 | 3 | 276 | 4762 |
| TC(Cluster 0)→PC | 32 | 26 | 1.230769 | 2.47619 | 7 | 2.461538 | 0.049231 | 3 | 212 | 3089 |
| TC(Cluster 1)→PC | 32 | 26 | 1.230769 | 2.47619 | 7 | 2.461538 | 0.049231 | 3 | 86 | 1673 |

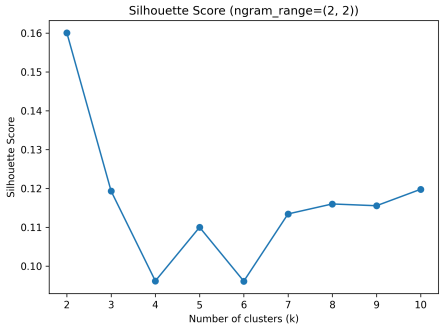


Fig. 2. Silhouette Score for different cluster sizes ($k$) using 2-gram features

cluster. On the other hand, the box plot method led to clusters with 1,045 and only 5 traces. This highly skewed distribution suggested that the box plot method was less effective in producing meaningful clustering results. Therefore, IQR-based feature selection was used for the subsequent experiment.

Among the materialized cells generated from the Process Cube, the (2014, 'Senior', 'Test') cell had the highest number of events, with a total of 4762, making it the most suitable choice for process discovery and subsequent evaluation. For the combination of TC→PC, the same Process Cube cell (2014, Senior, Test) was materialized, since it also contained the highest number of events within the clustered logs. This helps maintain consistent experimental conditions.

## 4.2 Model Complexity Results

To assess the impact of segmentation strategies, we evaluate four configurations: TC, PC, PC→TC, and TC→PC. Key metrics include CNC, CN, and Density. The remaining metrics such as P/T-CD, ACD, and others are summarized in Tables 1 and 2. The visual representations of the discovered process models for each segmentation method are provided in Appendix A.

*4.2.1 Trace Clustering (TC).* Cluster 0 showed a CNC of 1.31, CN of 29, and Density of 0.015. This indicates that although the process model exhibits relatively high CN, the overall densities of the total arcs and arcs between nodes are relatively low. Even though Cluster 1 has a lower number of traces, which is 3975, the result still shows a relatively high CN of 25. This suggests that clustering alone does not sufficiently reduce complexity.

*4.2.2 Process Cube (PC).* Within the materialized cells with dimensions of age, year, and activity, the (2014, Senior, Test) cell had a CNC of 1.223, CN of 8, and density of 0.041. It showed a simpler process model compared to those generated from trace clustering. This reveals that segmentation using the Process Cube is more effective than applying trace clustering alone.

*4.2.3 PC→TC.* This combination approach produced the lowest CN, with a value of 5, and the highest density at 0.083 in Cluster B. Cluster B also had the fewest number of events, with only 444, which helps explain the simplicity, since fewer events tend to have fewer variants. Cluster A has the lowest values in P/T-CD, ACD and CFC, while still maintaining ten times as many events as Cluster B. These values imply that this segmentation discovered the most simplified model among all segmentations.

*4.2.4 TC→PC.* When Process Cube was applied after clustering, results were similar to PC alone, since CN ranges between 7 and 8. However, a smaller sub-log from Cluster 1→PC had only 1673 events, which could limit discovery quality. Despite the smaller size, Cluster 1→PC and Cluster 0→PC showed very similar results across many metrics, including the number of arcs, number of nodes, CNC, P/T-CD, CN, ACD, Density, and CFC. These two combinations of segmentation showed almost the same result as the other combination of PC→TC (Cluster A).

## 5 DISCUSSION

In the results obtained from applying trace clustering alone, although the number of events differs significantly between the two clusters, the overall metric values do not show a considerable difference, except in the number of variants. This is likely due to the substantial difference in event counts between the clusters, with one having about four times more events than the other. Since the difference in the number of traces also shows big difference, most of the events are concentrated in the traces of Cluster 0, which likely played a key role in this result.

The primary reason that the process model generated through Process Cube segmentation had lower model complexity than those generated by trace clustering might have been due to its ability to navigate the segmentation direction and purpose more deliberately. Because the Process Cube allows event data to be divided through meaningful operational dimensions, specific cells can be extracted that share a similar behavioral pattern. As a result, the process discovery from such a cell is more likely to be consistent.

In the case of PC→TC combination, the density values were 0.04 and 0.08 for the two clusters, which are twice as dense compared to the results from Cluster 0 and Cluster 1. It could have been caused by the smaller number of events in each cluster. However, because Cluster A contains more events than Cluster 1, such an assumption cannot be substantiated. This suggests that the density increase cannot be explained solely by the event count. Moreover, other metric values showed simplification in general, and both the number of arcs and nodes decreased significantly. Therefore, the increased density in this combination does not always indicate a more complex overall process model.

The similarity between PC→TC (Cluster A) and TC→PC is also observed in the results from applying Process Cube alone. This may support the conclusion that the combination method is not significantly different from the outcome of Process Cube segmentation alone.

In the case of TC→PC combination, the overall results showed little difference from applying the Process Cube alone. When the Process Cube was applied to the clusters, the CNC, P/T-CD, ACD, and Density were slightly higher. This may be due to the smaller number of events in Cluster 1, but the results from Cluster 0→PC were not necessarily better than those of PC alone. Since the metric values are nearly identical but the event counts are lower, this may indicate that excessive fragmentation of the data can hinder analysis and adds complexity to interpretation.

### 5.1 Limitations

This study has several limitations. The dataset was confined to a single domain, and only basic dimensions such as age, time, activity were considered. Feature generation for trace clustering has been done with only n-grams. Additionally, performance metrics such as fitness and precision were excluded due to computational cost, which might have offered more process mining-centered insights into model quality.

## 6 CONCLUSION

This study conducted a benchmark of trace clustering and Process Cube-based segmentation in the context of process mining. The 4TU Sepsis Dataset is used to evaluate the process models discovered from four segmentation strategies : trace clustering , Process Cube, trace clustering → Process Cube, and Process Cube → trace clustering. The study aimed to investigate their impacts on model complexity and interpretability.

The results of different segmentation showed that the Process Cube → trace clustering combination had lowest values for Cyclomatic Number, Place Transition Connection Degree, Average Connected Degree, Control Flow Complexity. There were some segmentations that had not much of deviations between each other : Process Cube(2014, 'Senior', 'Test'), trace clustering(Cluster0) → Process Cube, and Process Cube → trace clustering(ClusterA). indicate that the Process Cube approach, either alone or in combination (especially Process Cube → trace clustering). The Process Cube → trace clustering method especially achieved the lowest cyclomatic number and the most reduced number of nodes and arcs.

Furthermore, the investigation reveals that the order of applying segmentation methods matters. Applying the Process Cube before clustering leads to having concise process models with events that have similar behavior, while trace clustering → Process Cube did not show a significant improvement in terms of complexity. Materialized cell data did not differ much from the clustered data. This suggests that dimension-based segmentation provides a meaningful initial grouping that can enhance the effectiveness of subsequent clustering.

However, while the findings are promising, they are based on a single dataset and limited dimensions, and this limits the applicability of the findings. Future work can include more diverse datasets, advanced feature engineering techniques with domain experts. Currently, dimensions of the process cube do not require domain-specific context, but inclusion of context-aware dimensions will bring meaningful outcomes for industries. The study has used only n-grams for feature generation, but active clustering [9] or conserved patterns [8] could be extracted for more elaborate pattern finding. Additional evaluation metrics such as fitness, precision, and f-score will offer new perspectives on analyzing the segmented data. Since evaluation of the study mainly focused on complexity of the process models, readability and understandability of the models and applicability of the segmentation method can provide more insights on future work.

## REFERENCES


[1] Wil van der Aalst. *Process Mining: Data Science in Action.* 2nd. Springer Publishing Company, Incorporated, 2016. ISBN: 3662498502. DOI: 10.1007/978-3-662-49851-4.

[2] Wil M. P. van der Aalst. "Process Cubes: Slicing, Dicing, Rolling Up and Drilling Down Event Data for Process Mining". In: *Asia-Pacific Business Process Management.* Ed. by Minseok Song, Moe Thandar Wynn, and Jianxun Liu. Vol. 159. Lecture Notes in Business Information Processing. Cham: Springer International Publishing, 2013, pp. 1–22. DOI: 10.1007/978-3-319-02922-1_10.

[3] Annalisa Appice and Donato Malerba. "A Co-Training Strategy for Multiple View Clustering in Process Mining". In: *IEEE Transactions on Services Computing* 9.6 (2016), pp. 832–845. DOI: 10.1109/TSC.2015.2430327.

[4] Annalisa Appice and Donato Malerba. "A Co-Training Strategy for Multiple View Clustering in Process Mining". In: *IEEE Transactions on Services Computing* 9.6 (2016), pp. 832–845. DOI: 10.1109/TSC.2015.2430327.

[5] Till Becker and Wacharawan Intoyoad. "Context Aware Process Mining in Logistics". In: *Procedia CIRP* 63 (2017), pp. 557–562. DOI: 10.1016/j.procir.2017.03.149.

[6] Antonina K. Begicheva, Irina A. Lomazova, and Roman A. Nesterov. *Discovering Hierarchical Process Models: an Approach Based on Events Clustering.* 2023. DOI: https://arxiv.org/abs/2303.13531. arXiv: 2303.13531 [cs.AI].

[7] Alfredo Bolt and Wil M. P. van der Aalst. "Multidimensional Process Mining Using Process Cubes". In: *Enterprise, Business-Process and Information Systems Modeling.* Ed. by Khaled Gaaloul et al. Vol. 214. Lecture Notes in Business Information Processing. Cham: Springer International Publishing, 2015, pp. 102–116. DOI: 10.1007/978-3-319-19237-6_7.

[8] R. P. Jagadeesh Chandra Bose and Wil M. P. van der Aalst. "Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models". In: *Business Process Management Workshops*. Ed. by Stefanie Rinderle-Ma, Shazia Sadiq, and Frank Leymann. Vol. 43. Lecture Notes in Business Information Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 170–181. DOI: 10.1007/978-3-642-12186-9_16.

[9] Jochen De Weerdt et al. "Active Trace Clustering for Improved Process Discovery". In: *IEEE Transactions on Knowledge and Data Engineering* 25.12 (2013), pp. 2708–2720. DOI: 10.1109/TKDE.2013.64.

[10] Maikel L. van Eck et al. "PM$^2$ : *AProcessMiningProjectMethodology*". In: *Advanced Information Systems Engineering*. Ed. by Jelena Zdravkovic, Marite Kirikova, and Paul Johannesson. Vol. 9097. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 297–313. DOI: 10.1007/978-3-319-19069-3_19.

[11] Yutika Amelia Effendi and Minsoo Kim. "Reliable Process Tracking Under Incomplete Event Logs Using Timed Genetic-Inductive Process Mining". In: *Systems* 13.4 (2025). DOI: 10.3390/systems13040229.

[12] Nik F Farid, Marc de Kamps, and Owen A Johnson. "A Process Cube based Approach of Process Mining in Analysing Frailty Progression Exploiting Electronic Frailty Index." In: *HEALTHINF*. 2022, pp. 605–613. DOI: https://doi.org/10.5220/0010879200003123.

[13] Anahita Farhang Ghahfarokhi, Alessandro Berti, and Wil M. P. van der Aalst. *Process Comparison Using Object-Centric Process Cubes*. 2021. DOI: https://arxiv.org/abs/2103.07184. arXiv: 2103.07184 [cs.DB].

[14] G. Greco et al. "Discovering expressive process models by clustering log traces". In: *IEEE Transactions on Knowledge and Data Engineering* 18.8 (2006), pp. 1010–1027. DOI: 10.1109/TKDE.2006.123.

[15] Mohammad Imran et al. "Complex Process Modeling in Process Mining: A Systematic Review". In: *IEEE Access* 10 (2022), pp. 101515–101536. DOI: 10.1109/ACCESS.2022.3208231.

[16] Amin Jalali. "Object Type Clustering Using Markov Directly-Follow Multigraph in Object-Centric Process Mining". In: *IEEE Access* 10 (2022), pp. 126569–126579. DOI: 10.1109/ACCESS.2022.3226573.

[17] Pieter De Koninck et al. "Expert-driven trace clustering with instance-level constraints". In: *Knowledge and Information Systems* 63 (2021), pp. 1197–1220. DOI: 10.1007/s10115-021-01548-6.

[18] Phuong Nguyen et al. "Process Trace Clustering: A Heterogeneous Information Network Approach". In: *Proceedings of the 2016 SIAM International Conference on Data Mining (SDM)*. Society for Industrial and Applied Mathematics, 2016, pp. 279–287. DOI: 10.1137/1.9781611974348.32.

[19] Fareed Zandkarimi et al. "A Generic Framework for Trace Clustering in Process Mining". In: *2020 2nd International Conference on Process Mining (ICPM)*. IEEE, 2020, pp. 177–184. DOI: 10.1109/ICPM49681.2020.00034.

# A DISCOVERED PROCESS MODELS FOR EACH SEGMENTATION METHOD

This appendix provides the visualization of the discovered process models using four different segmentation methods. Each model was generated through Inductive Miner using PM4Py in Python. The key metrics of these models are discussed in section 4.2.

## A.1 Trace Clustering (TC)

**Cluster 0.** This cluster contains a large number of events with a relatively complex process model. It shows numerous behavioral variants and several parallel branches.

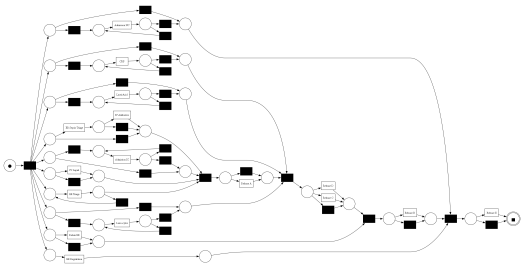

Fig. 3. Discovered process model for TC(Cluster 0)

**Cluster 1.** This cluster includes fewer traces and exhibits a much simpler process model with fewer activity transitions.

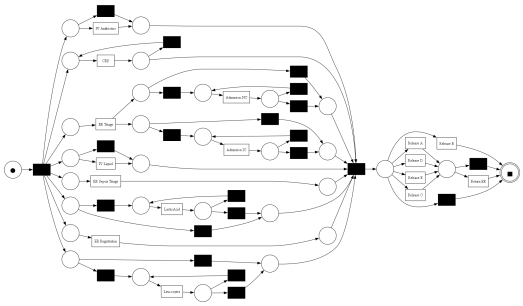

Fig. 4. Discovered process model for TC(Cluster 1)

## A.2 Process Cube (PC)

**Cell (2014, 'Senior', 'Test').** This cell represents a subset of the event log filtered by the Process Cube based on the dimensions: age, year, and activity. The discovered process model is notably simpler than those generated from clustering, with fewer variants and a clearer control-flow structure.

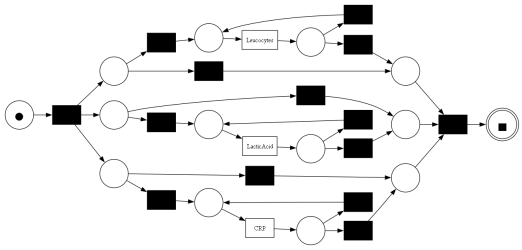


Fig. 5. Discovered process model for PC - Cell (2014, Senior, Test)

## A.3 PC → TC

**Cluster A.** This cluster is obtained by first applying the Process Cube to segment the data based on contextual dimensions, and then performing trace clustering on the most populated cell. The resulting process model is significantly simplified, with fewer arcs and nodes.

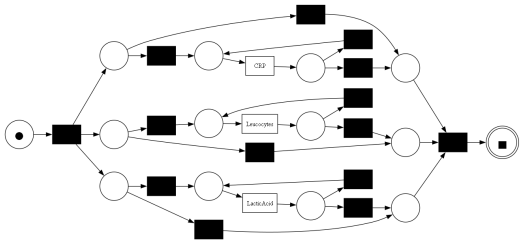


Fig. 6. Discovered process model for PC → TC(Cluster A)

**Cluster B.** This cluster contains fewer events and traces, resulting in an even simpler process model. It has the highest arc density but the lowest complexity in terms of process structure.

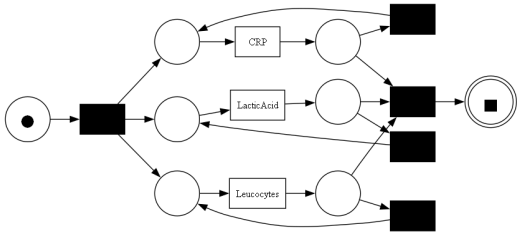


Fig. 7. Discovered process model for PC → TC(Cluster B)

## A.4 TC → PC

**Cluster 0 → PC.** The Process Cube is applied to the Cluster 0 from the trace clustering results. The resulting model shows a moderate level of complexity, similar to that of PC → TC (Cluster A), indicating that both segmentation orders can lead to comparable simplification in some cases.

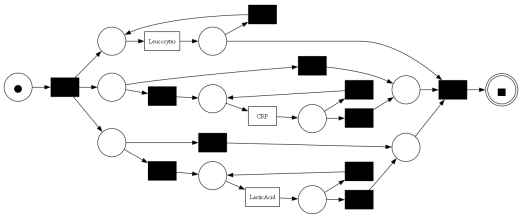


Fig. 8. Discovered process model for TC(Cluster 0) → PC

**Cluster 1 → PC.** Applying the Process Cube to Cluster 1 produces a simpler process model. However, due to the smaller volume of data, it may not fully capture the behavioral diversity. The model has fewer variants and lower complexity metrics.

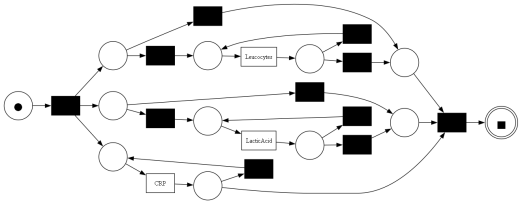


Fig. 9. Discovered process model for TC(Cluster 1) → PC