SEQUENCE ALIGNMENT BASED PROCESS FAMILY EXTRACTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

EREN ESGİN

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

DECEMBER 2018

SEQUENCE ALIGNMENT BASED PROCESS FAMILY EXTRACTION

Submitted by **EREN ESGİN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in the Department of Information Systems, Middle East Technical University** by,

Prof. Dr. Deniz ZEYREK BOZŞAHİN
Dean, **Graduate School of Informatics** _____

Prof. Dr. Yasemin YARDIMCI ÇETİN
Head of Department, **Information Systems** _____

Prof. Dr. Pınar KARAGÖZ
Supervisor, **Computer Engineering, METU** _____

Prof. Dr. Yasemin YARDIMCI ÇETİN
Co-Supervisor, **Information Systems, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Aysu BETİN CAN
Information Systems, METU _____

Prof. Dr. Pınar KARAGÖZ
Computer Engineering, METU _____

Assoc. Prof. Dr. Osman ABUL
Computer Engineering, TOBB ETU _____

Assoc. Prof. Dr. Erhan EREN
Information Systems, METU _____

Asst. Prof. Dr. Banu AYSOLMAZ
Information Management, Maastricht University _____

Date: 17.12.2018

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Eren ESGİN

Signature : _____

ABSTRACT

SEQUENCE ALIGNMENT BASED PROCESS FAMILY EXTRACTION

Esgin, Eren
Ph.D., Department of Information Systems
Supervisor: Prof. Dr. Pınar Karagöz
Co-Supervisor: Prof. Dr. Yasemin Yardımcı Çetin

December 2018, 190 pages

Business Process Management (BPM) gains growing attention by generic process design and execution capabilities empowered by process-aware information systems. During execution of these transactional information systems, end-users leave traces in the form of event logs, which can be used as a main data source for behavior analysis. Process mining encompasses the techniques for automatically discovering process from these event logs, checking conformance between the reference process model and process executions, as well as analyzing, predicting and enhancing the performance of business processes. With the emergence of new shared economical models and system architectures, monolithic process perspective is evolved through cross-organizational applications. While contemporary information systems provide functionality for process management within the organizations, a systematic approach to support and analyze multi-organizational processes is missing. Cross-organizational process mining supports the use of commonality and collaboration for process configuration. However, this functionality creates the challenge of dealing with variability across organizations.

In this study, we propose a three phased cross-organizational process mining framework in order to extract the commonalities among different organizations serving the same business values. While dominant behavior extraction phase initially derives the sequence of tasks expressing the most typical behavior within the process instances, sequence alignment phase measures the degree of similarities between the process candidates by confidence enhanced cost functioning, and depicts the neighborhood among these alternatives in terms of process families. At process configuration phase, common regions that indicate a functional inheritance or abstractions in the process families are visualized at sequence alignment matrices and interpreted by new feature sets, namely identical and maximal identical pair. According to the experimental results, proposed approach presents a viable and robust cost function in incorporating the business context at process similarity measurement and clustering the process alternatives into process families.

Keywords: Cross-Organizational Process Mining, Process Families, Multi-Sequence and Pairwise Alignment, Dominant Behavior, Identical and Maximal Identical Pairs.

ÖZ

DİZİ HİZALAMA BAZLI SÜREÇ AİLELERİNİN ÇIKARIMI

Esgin, Eren
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Prof. Dr. Pınar Karagöz
Tez Yardımcı Yöneticisi: Prof. Dr. Yasemin Yardımcı Çetin

Aralık 2018, 190 sayfa

Süreç duyarlı bilgi sistemleri tarafından iyileştirilen genel süreç tasarımı ve yürütme işlevleri ile İş Süreçleri Yönetimi alanı artan bir ilgili toplamaktadır. Bu bilgi sistemlerinin yürütülmesi sırasında elde edilen olay günlükleri, son kullanıcı davranış analizlerinde ana veri kaynağı olarak kullanılabilir. Süreç madenciliği, bu günlüklerden iş süreçlerinin otomatik keşfedilmesi ve referans süreç modeli ile süreç gerçekleştirimi arasında uygunluk kontrolünün yapılmasının yanı sıra, iş süreci performanslarının analizi, tahmini ve geliştirilmesini de kapsar. Yeni paylaşım ekonomisi modelleri ve sistem mimarilerinin ortaya çıkışıyla, tekil süreç perspektifi organizasyonlar arası uygulamalara doğru evrilmiştir. Güncel bilgi sistemleri, organizasyonel bağlamda süreç yönetimi için işlevsellik sunarken, çoklu-organizasyonel süreçleri desteklemek ve analiz etmek için gerekli sistematik yaklaşımdan uzaktır. Organizasyonlar arası süreç madenciliği, süreç yapılandırması için benzerliğin ve işbirliğinin kullanımını destekler. Bununla birlikte, bu işlevsellik organizasyonlar arasında ortaya çıkan değişkenliklere çözüm bulma zorunluluğunu da yaratır.

Bu çalışmada aynı süreçlere odaklanan farklı organizasyonlar arasındaki benzerlikleri ortaya çıkarmak için üç aşamalı bir organizasyonlar arası süreç madenciliği çerçevesi sunuyoruz. Baskın davranışın çıkarımı aşaması, süreç örneklerinde gözlemlenen en tipik davranışı ifade eden görev dizisini türetirken, dizi hizalama aşamasında ise güvene dayalı maliyet işlevine göre süreç alternatifleri arasındaki benzerlik derecesi ölçülüp, ilgili süreç alternatifleri arasındaki komşuluklar süreç aileleri üzerinden görselleştirilir. Son olarak süreç yapılandırılması aşamasında, süreç ailelerinde tespit edilen fonksiyonel benzerlikler ve soyutlamalar dizi hizalama matrislerinde görselleştirilip, özdeş veya azami özdeş çiftleriyle yorumlanır. Deney sonuçlarına göre önerilen yaklaşım, süreç bağlamına göre benzerlik ölçümünde ve süreç alternatiflerinin kümelenmesinde uygulanabilir ve sağlam bir benzerlik ölçümü sunmaktadır.

Anahtar Sözcükler: Organizasyonlar Arası Süreç Madenciliği, Süreç Aileleri, Çok Sıralı ve Çift Yönlü Hizalama, Baskın Davranış, Özdeş ve Azami Özdeş Çiftler.

babama, ruhu şad olsun…

ACKNOWLEDGEMENT

First of all, I would like to express my sincere gratitude to my supervisor Prof. Dr. Pınar Karagöz for her continuous guidance, insightful suggestions and comments throughout my study. I also deeply appreciate her never ending patience and encouragements.

I particularly would like to thank to Prof. Dr. Yasemin Yardımcı Çetin for all her valuable helps, comments and contributions in this study.

I wish to express my appreciation to Assoc. Prof. Dr. Aysu Betin Can, Assoc. Prof. Dr. Osman Abul, Assoc. Prof. Dr. Erhan Eren and Asst. Prof. Dr. Banu Aysolmaz for their support, critical suggestions and valuable comments throughout the steering meetings and thesis committee.

I am thankful to my friends, Çağdaş Bayraktar, Dr. Deniz Akdur, Dr. Pınar Efe, Ali Aydoğan, Samet Karahacıoğlu, Aydın Erken, Engin Çark, Bilge Can Gürer, Hasan Onur Beygo, Duygu Küçükbahar Beygo, Seher Tavukçu Güldoğan, Devrim Derici and Ender Yalçın for their continuous encouragement, infinite tolerance and moral support in my never ending studies during years.

Last but not the least; I would like to thank my family for supporting me spiritually throughout my life. Special thanks go to my dear mother Ürkiye Esgin; to my beloved sister Esma Esgin Günder, her kind husband İzzet Günder and my little nephew Gülce Günder for their encouragement and support throughout this academic journey. I am really happy and fortunate to be a part of this wonderful family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

AHC	:	Agglomerative Hierarchical Clustering
APROMORE	:	Advanced Process Analytics Platform
BI	:	Business Intelligence
BPI	:	Business Process Intelligence
BPM	:	Business Process Management
BPMN	:	Business Process Modeling Notations
CANW	:	Confidence-aware Needleman-Wunsch
CCA	:	Cost Center Accounting
CM	:	Causality Metric
COGS	:	Cost of Goods Sold
ConfEnhMSA	:	Confidence Enhance Multi-Sequence Alignment
confFTC	:	Confidence for From-To Chart
cosSim	:	Cosine Similarity
CRM	:	Customer Relationship Management
EMiT	:	Enhanced Mining Tool
EPC	:	Event Process Chain
ERP	:	Enterprise Resource Planning
FLP	:	Facility Layout Problem
GA	:	Genetic Algorithms
GM	:	Global Metric
HMM	:	Hidden Markov Models
InWoLvE	:	Inductive Workflow Learning Via Examples
IP	:	Integer Programming
IP	:	Identical Pair
IR	:	Information Retrieval
IT	:	Information Technology
KPI	:	Key Performance Indicators
LCSS	:	Longest Common Subsequence
LM	:	Local Metric
LP	:	Linear Program
maxIP	:	Maximal Identical Pair
ML	:	Modified Lift
MSA	:	Multi-Sequence Alignment
NW	:	Needleman-Wunsch
PA	:	Pairwise Alignment
PAIS	:	Process-Aware Information Systems
PC	:	Product Costing
PDM	:	Product Data Management
PTS	:	Principal Transition Sequences
QAP	:	Quadratic Assignment Problem
SaaS	:	Software as a Service
SAP	:	Systems, Applications and Product
SBPMI	:	Shared Business Process Management Infrastructures
SCM	:	Supply Chain Management

simScr	:	Similarity Score
SNA	:	Social Network Analysis
SOX	:	Sarbanes-Oxley
SP	:	Sum-of-Pairs
SSD	:	Semantic Similarity Degree
suppFTC	:	Support for From-To Chart
TAR	:	Transition Adjacency Relations
TWLCS	:	Time-Wrapped Longest Common Subsequence
WFMS	:	Workflow Management Systems

CHAPTER 1

INTRODUCTION

The tendency towards more open economies has enforced the evolution of *globalization*, which suggests that the world is a single broad market that can be accessed by all industries and organizations. The transformations that happened in the markets are so rapid and volatile that only the most flexible and agile form of organizations can adapt to these trends of change. Indeed, ultimate action at this enterprise transformation is to achieve a holistic and sustainable business process management (BPM) with adequate key performance indicators (KPI's). The process orchestration throughout the value chain of the underlying organizations is possible by the *process-aware information systems*, which are information systems that manage and orchestrate major business processes at organizations. While these information systems are intensively implemented, their business value and functionality they provide are limited due to how processes are traditionally designed [1]. Unfortunately, process design is influenced by the personal perceptions and reference process models are mostly normative such that they reflect what should be done rather than the actual process executions. Consequently, process design tends to be incomplete, subjective and at a coarse-grained level [2, 3]. Actually, major problems emerge because processes are actually performed differently than they are designed [3, 4].

Process mining is anticipated as a solution to handle these limitations by distilling end-user behavior patterns from event logs and discovers the process knowledge [1, 4, 5, 6, 7, 8]. It encompasses the techniques of discovering processes automatically, checking the conformance between the reference process model and process executions, as well as analyzing, predicting and enhancing the performance of business processes [9]. Event logs reflect what the process owners or end users perform at the operational level. Thus, unlike the traditional design-centric approach, process mining is not biased by subjective perceptions [10].

With the emergence of new shared economy models and information system architectures, e.g. shared BPM infrastructures and cloud computing [11], the scope of process-aware information systems is extended towards *cross-organizational applications*. While these contemporary information systems are intensively utilized at organizations, they serve a limited functionality to fulfill the business requirements of multi-organizational processes [12, 13]. For instance, enterprise resource planning (ERP) systems focus on specific functionalities determined for an exact purpose [12, 14]. These information systems are configured through a time-consuming customization phase and these customizations are relatively *data-centric* and relatively complex, i.e. processes are hindered at the application tables [13]. Moreover, large installment core acts as an inhibiting factor that complicates the software refactoring and *process-centric* transformation [13]. Multi-tenant processes, i.e. organizations executing the same processes at a shared or distributed architecture, require a more systematic treatment to deal with the variability across the organizations. The requirement for such information systems makes the *monolithic* perspective of former process mining techniques evolve through a new era, namely *cross-organizational process mining*. This new type of process mining focuses on *exploiting the commonalities* between the organizations, and *collaboration* [11, 15]. These settings are fundamental for *configurable process models*, which provide generic structures representing possible variations of a process in an integrated, single process model [16]. To build up configurable process models by integrating different process variants, first, they need to be compared by measuring their similarity and deviations.

While process similarity measurement is hindered by different modeling notations, task labeling styles and terminology, process similarity has been measured by three complementary aspects: the task labels, dependencies between the tasks and the process semantics [17, 18]. Current techniques have most measured the similarity between process models, in other words *model-model similarity*, based on a semantic and syntactic comparison of task labels and process models together [21]. This makes it feasible to adapt algorithms from information retrieval (IR) and graph theory for measuring process similarity [19]. However, such adaptations have been found to be inadequate to take the process behavior into account, for example, when two process variations look similar in terms of task labels or process structure, but may behave quite distinct [19]. As an alternative, *log-model* similarity measurement uses the behavior of a process model by instantiating the state space or enumerating all possible traces by implementing trace equivalence, bisimulation or branch simulation techniques [20]. Although various similarity measures overcome potential scalability problems emerged by trace enumarations and reflect the process branchings at process behavior in a polynomial time, they majorly promise a limited *binary (true/false) similarity response* instead of the similarity degree [20, 21]. Additionally, they do not assign priorities (or weights) to the sub-processes according to their execution frequencies [20]. Although process mining has overwhelmed various problems encountered at handling real life use cases, there exist still challenging issues that should be handled in the context of process mining applications on event logs and one of these topics is *process diagnostics*, i.e. measuring the compliance between reference and actual models, interpreting for related process variants at organization repositories [19, 21]. In both model-model and log-model similarity measurement, diagnostics of processes at the model level is time-consuming and sometimes infeasible, especially when dealing with flexible processes delimited by concept drift, i.e. the characteristics of underlying process alter over time [21]. Respectively, similarity measurement on the basis of process execution semantics, in other words log-log similarity measurement, bypasses the requirement of such reference process models.

Due to these limitations observed in current process similarity metrics at cross-organizational applications, we propose a *cross-organizational process mining framework* for extracting the commonalities among different organizations serving the same business values. For this purpose, we aim to segregate the organizations into process clusters, in other words *process families*, by measuring the similarity according to process executions. As shown in Figure 1.1, the underlying framework consists of three phases: *dominant behavior extraction*, *sequence alignment* and *process configuration*.

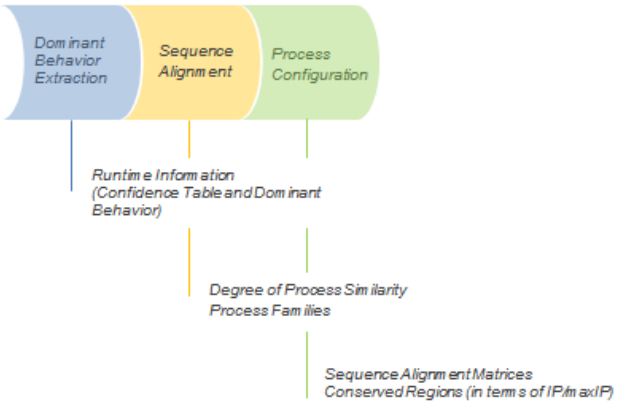


Figure 1.1. The Overview for Proposed Three-Phased Cross-Organizational Process Mining Framework and Major Outcomes.

Dominant behavior extraction phase initially derives the representative sequence that decodes the *dominant behavior*, i.e. a typical or common intended behavior that can act as the backbone of the underlying process. Unlike the model-model and log-model similarity measurements, this exemplary behavior acts an appropriate abstraction for the corresponding process behavior by eliminating the ultimate requirement for well-defined reference process modeling. In addition to the dominant behavior, this phase summarizes the *confidence values*, which are the intra-dependencies of consecutive task pairs sharing an incorporated business context. Accordingly, this confidence notion reflects the moment of choice at the process behavior and provides insights about the level of inter-dependencies between the tasks or activities.

The following phase, sequence alignment employs the adaptations of Needleman-Wunsch (NW) algorithm, which basically proposes a dynamic programming (DP) paradigm to find out the optimal alignment between two amino acid sequences [22]. Indeed, sequence alignment is an essential tool in Bioinformatics domain to identify the similarity between two biological sequences to understand their structures or functions [22, 23]. Our major motivation of sequence alignment phase is to measure the degree of similarities between the process alternatives according to the overlapping regions that are detected by two sequence alignment adaptations: *Multi-Sequence Alignment* (MSA) and *Pairwise Alignment* (PA). Primarily, MSA depicts the commonalities in terms of *process family tree*, which is a dendrogram-like guide tree that progressively captures the distance among the process alternatives. As the former NW adaptation, PA segregates the process alternatives by various clustering algorithms in terms of similarity scores as the distance attributes. However, there may emerge various challenges in sequence alignment adaptation such as determining the cost function at edit operations. While BOM and BLOSUM pay-off matrices are frequently used in Bioinformatics literature [22], we develop a dynamic cost functioning based on the confidence values obtained at dominant behavior extraction phase. The fundamental motivation of *confidence enhanced cost functioning* is to eliminate the edit operations that contradict with the underlying business context: while the substitution of contrasting activities and inDel (insertion/deletion) operations of activities with little compatibility for the corresponding business rules should be avoided by dynamically determined penalty scores, the tasks with complementary business circumstance should be encouraged to be substituted or inserted at practical costs.

As the final phase, process configuration visualizes the sequence alignments among the process alternatives that are assigned to the same process family. The deviations and exceptional process behaviors are emphasized by the regions that are rarely filled with gap symbol (–), as emphasized in [21], conserved regions that are detected by the *identical* and *maximal identical pairs* feature sets (IP and maxIP) emphasize a functional inheritance among the underlying process alternatives. Consequently, these conserved regions can be used to create various abstractions at a desirable level of granularity for configurable process modeling and the divergences across the process alternatives can be dealt with configurable elements.

The major contributions of the proposed cross-organizational process mining framework are as follows:

- The approach focuses on the sequential dominant behavior of process alternatives. In this way, the requirement for the existence of a reference process model is relaxed, which is a common limitation in current approaches [20, 21].
- Sequence alignment techniques have been applied as a preprocessing step on the event logs in the process mining literature [21, 23, 34, 60, 61, 62]. As a distinction in this study, we adapt sequence alignment on process model variants of the same process in order to measure the degree of similarity on a continuous scale, instead of a limited atomic similarity response.
- This work is the first to adapt NW algorithm with robust cost function to construct process clusters that highlight the major commonalities among the process alternatives. This cost function relies on the business context such that, edit operations are dynamically valuated according to the compliance of operation to the corresponding business rules.
- As the business value, the alignments of process alternatives that are assigned to the same process family can play a significant role in process configuration such that, conserved regions detected by maximal identical pairs (maxIP) with higher frequency and coverage are interpreted as an evidence of common behavior and manifestation of these concurrent behaviors highlight a functional inheritance at process enactment.

This study is composed of seven chapters. Enterprise transformations and paradigm shifts observed in process mining research area are analyzed in Chapter 2. Prior aspects and approaches in process discovery, process similarity measurement and process configuration fields are summarized in Chapter 3. Chapter 4 highlights the background information for former concepts that are intensively addressed at following sections. The details about the proposed three phased cross-organizational process mining framework are given in Chapter 5. Experimental analyses of the proposed framework with respect to four distinct use cases are handled in Chapter 6. Finally, the limitations and suggestions about the future work are explained and the concluding remarks are summarized in Chapter 7.

CHAPTER 2

PROBLEM ANALYSIS

2.1. Globalization and Enterprise Transformation

Due to the *globalization*, organizations face serious challenges that enforce rapid and sustainable enterprise transformation. This transformation implies the strategical business agility to respond to the competitors' reactions (e.g. new competitor or product intrusions to the market) and the ability to predict new opportunities at the market. Indeed, this transformation has a direct effect on the Business Process Management (BPM) such that, there has emerged a paradigm shift from *data-oriented* towards *process-oriented* organizational structures. The degree of this enterprise transformation may vary from Business Process Intelligence (BPI), which is a common key word for the techniques under the Business Intelligence (BI) technology [8], to the paradigm shift in the processes supported by the organization.

Actually managing critical business processes seek the development of contemporary information systems with the capability of monitoring and supporting the corresponding business processes. Such information systems are called *process-aware information systems* (PAIS) that offer generic process modeling and execution functionalities to bridge the perceived gap between the organization and the software by controlling and monitoring the information flow [24]. Enterprise Resource Planning (ERP), Workflow Management Systems (WFMS), Customer Relationship Management (CRM), Supply Chain Management (SCM) and Product Data Management (PDM) software can be classified as process-aware information systems [1, 6].

Despite process-aware information systems promise management of the tasks, it is delimited with a set of fundamental problems that result in critical barriers at practical use. Major drawback of such an information system is that the reference process models generated at process design phase lead to a lack of flexibility, which means an incapability to transform the processes without loss of any identity and functionality [1]. Indeed, process design phase is often orchestrated by a small group of consultants, process observers and domain experts and these stakeholders state what should be done rather than describing the actual business process [2, 10]. As shown in Figure 2.1, traditional process design majorly concentrates on the design and configuration phases, which are dominated by the managerial ideas on refining the business practices. Consequently, there happens a representation gap between process design and process enactment [7] and the final design is often incomplete, subjective and at a too high level [1, 2].

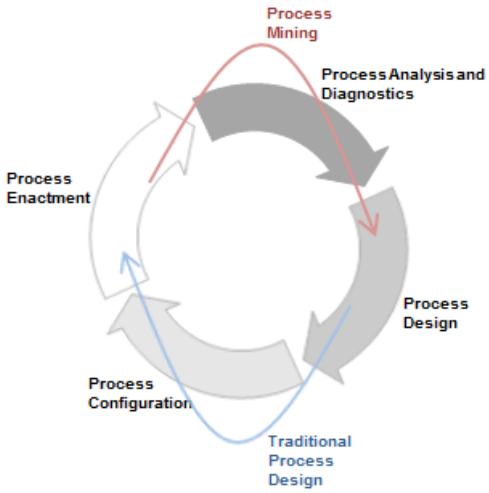


Figure 2.1. Traditional Process Design Life Cycle versus Process Mining [10].

Moreover, the flexibility notion denotes the ability to yield to change without loss of identity and business process flexibility is the capability to react to the external changes (e.g. *concept drift*) by adapting the sub-processes that are affected and required to be revised [1]. However traditional process design is inflexible to these changes, due to the strong *push-oriented nature* of routing, which imposes what to do instead of letting a free choice to the process observers [1]. This push-oriented nature of process design results in *context tunneling*, i.e. the end-users have no overview or a holistic idea about the underlying process [10]. This phenomenon is also in parallel with the *scientific management* and *standardization of the work* ideas of Frederick W. Taylor to partition the work in order to make it easy-allocated. The causality in the enterprise transformation can be modeled as a reinforcement cycle as shown in Figure 2.2.

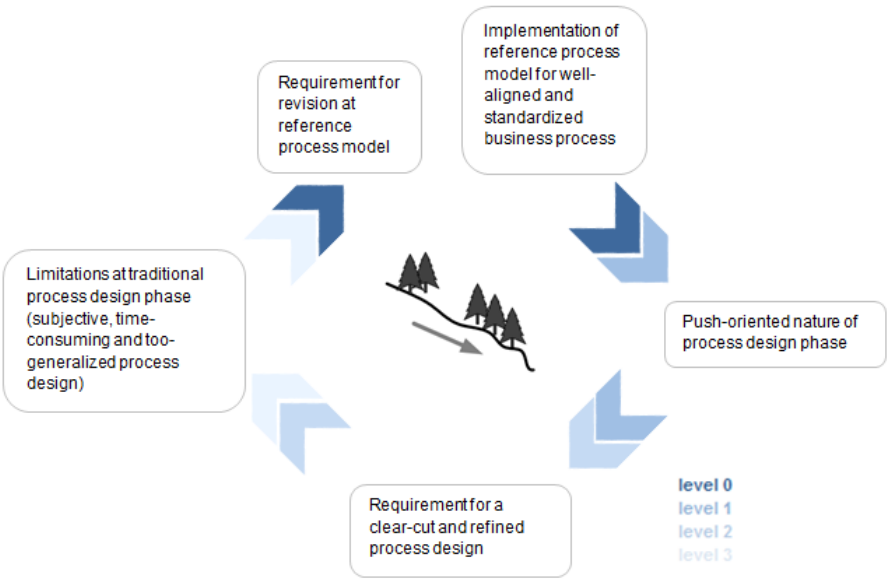


Figure 2.2. Reinforcement Cycle for Traditional Approach in Workflow Technology.

2.2. Process Mining

As stated above, the impact of process-aware information systems is limited by the difficulties encountered at the process design phase [7]. Respectively, reference process models are often normative in the sense that they reflect what should be done rather than the actual process execution [10]. This requires an expensive and time-consuming process analysis, which may be infeasible from an economical perspective. Instead of manually designing the process, it is proposed to reverse the underlying procedure by a more objective and automated way of design, which collects the process knowledge and discovers the underlying process patterns from this low-level process history called *event logs* [10].

The term *process mining* is concerned with this objective approach to discover, monitor and improve the real processes by distilling the process knowledge from event logs. Hence process mining describes a family of posteriori process models, which exploits end-user behaviors embedded at the event logs in contrast to the ideal picture at apriori reference process modeling [25] as shown in Figure 2.1. Indeed, process mining is unbiased towards the perceptions or normative decisions unlike traditional process design [6, 7]. However, if the process observers bypass the underlying process-aware information system by various alternative work-arounds that are quite different from the business rules and standard operations procedures, the event logs may deviate from the actual case [10]. Indeed, process mining is not an instrument to re-design the process models, it is better to compare the manually designed reference models with the discovered ones.

Respectively, the fundamental information at the event logs should cover the following attributes:

i. Each event refers to an *activity* (i.e. an atomic task in the process).
ii. Each event refers to a *case* (or *process instance*). Processes are by definition *case-based*, i.e. every piece of work is performed for a specific case [2].
iii. Each event can have an *originator* (i.e. the end-user executing or initiating the activity).
iv. Events have a *timestamp* and are totally ordered by case identifier.

The only assumption about process mining is the possibility to collect such a process history in terms of event logs. Process mining can be distinguished into three perspectives:

- *Process perspective*. The process perspective concentrates on the *control-flow* aspect, i.e. the ordering of the tasks. The major goal is to derive a good behavioral characterization from process executions [15].
- *Organizational perspective*. The organizational perspective concentrates on the originator attribute of the event logs. The goal is to figure the interactions between the process observers at the underlying organization by categorizing the process observers in terms of profiles or roles. The derived interactions are depicted by social network analysis (SNA).
- *Case perspective*. This perspective majorly focuses on the case features. Certainly, process instances are featured by the values of the corresponding data elements, e.g. the travel destination and advanced payment option of a travel request at travel management business process. Alternatively, this case perspective figures out the correlation between the activity occurrences and the process features.

Respectively, process mining is related to the process execution phase where much flexibility is potential such that, the more ways in which process observers deviate, the more variability is to be observed at end-user behavior analysis. In this aspect, there are three basic types of process mining:

a. *Process discovery*. The aim of process discovery is to extract information from the event logs in the form of process models. The forms of extracted process model vary such as event process chain (EPC) diagram, petri-nets, sociograms or time charts describing the process performance. Process discovery does not require a predefined apriori process model, but discovered process patterns can be used as the baseline at *delta analysis*, which compares these discovered process patterns that characterize actual process executions with the apriori process model [6, 7].
   The major challenge at process discovery is to convert extracted process patterns into valid process modeling notations. Additionally, this representation should avoid any spaghettiness that may increase the complexity of process discovery.

b. *Conformance checking*. Unlike the process discovery, conformance checking requires an apriori process model to compare observed process patterns and the to-be business process. Hence it is possible to perceive the discrepancies between process design and actual process behaviors by conformance checking. Additionally, the bottlenecks and rarely active process fragments can be detected.
   Rediscovery problem is a critical issue at conformance checking such that, the process mining algorithm is required to be able to extract a process model that is behaviorally equivalent to the reference process model, from which the complete event logs are generated [15].
c. *Extension*. Like conformance checking, extension requires an apriori process model which is enriched with new aspects obtained at process discovery. For instance, process mining applications can be implemented at ERP systems to simplify and improve the customizations steps [26].

Figure 2.3 summarizes the types of process mining.

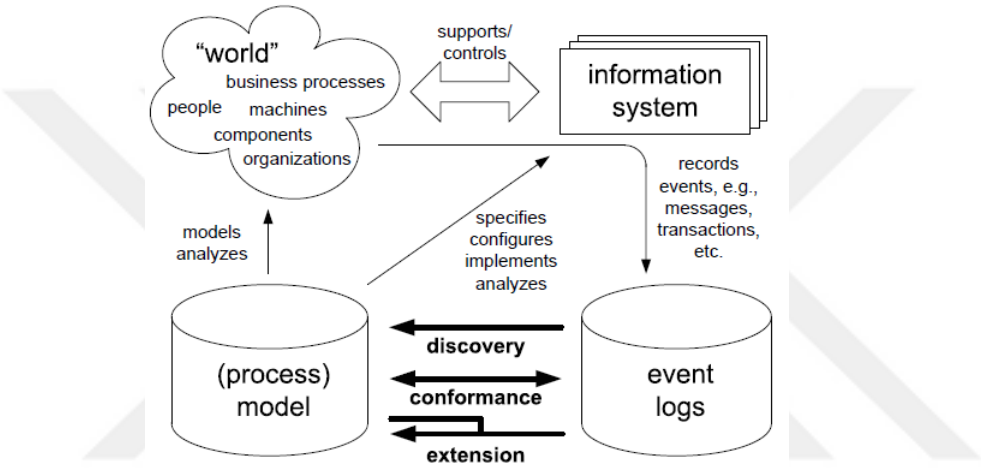


Figure 2.3. Types of Process Mining [25].

Beside the capabilities and functionalities provided by process mining, there are various challenges that are basically common at data mining domain:

- *Completeness*. Complex and spaghetti-like business processes may exhibit alternative or parallel patterns and the event logs may not typically reflect all of these possible process sequences and inter-leavings [10]. Moreover, certain process executions may have a low probability (i.e. *surprise-type* relations) and therefore remain undetected.
- *Noise*. Some parts of the event logs can be incorrect, incomplete or addressed to the process exceptions. These events can be distorted due to the human by-effect or various technical problems [10]. Also there may happen missing event logs, if some of the activities are performed manually or handled at an external system [10].

Increasingly, professional organizations are looking for the benefits of sharing their best practices among their stakeholders and this era is mostly called as *shared economy* [27]. While cloud computing focuses on the sharing of information technology (IT) investments and the assets to achieve significant cost reductions, software as a service (SaaS) is another complimentary notion which refers to a new software distribution model. It is a repository that hosts various applications from distinct vendors or service providers [12].

Since multi-tenant infrastructure also enables to hold the event logs of multiple organizations, *monolithic perspective* of traditional process mining (i.e. process discovery or conformance checking within a single organization) is evolved through *cross-organizational process mining*. This type of process mining handles the major similarities between the process structures and end-user behaviors throughout the cross-analysis and the challenges about horizontal or vertical segmentation of the tasks and business processes [28]. Although cross-organizational process mining promises various business values, there are major limitations for its implementation in cooperative organizations. These are the identification of the commonalities and discrepancies between the ways they actually work and the

integration of these possible process variations into a single adaptive model [27]. Instead of *dealing with the variability* over the organizations, enforcing the “one size fit all” aspect to all requirements and preferences is infeasible [12].

Due to the challenge of dealing with the variability across the organizations, two major settings are emphasized:

- *Collaboration*. Collaboration setting refers to the process instances handling of a distinct process distributed over different organizations. This option highlights the *interoperability* among different organizations. The corresponding process is analog to the “jigsaw puzzle” metaphor, i.e. the process is cut into loosely-coupled tasks [15]. This process fragmenting is also called as *horizontal partitioning*. The major challenge about this form of cross-organizational process management is the myopicism, which means the organizations only focus on a limited process fragment [15].
- *Exploiting commonality*. In this setting, the major goal is not to distribute the tasks associated to a business scenario. Instead, it is aimed to share the experiences, business knowledge or common best practices among the organizations executing essentially similar processes [15]. This setting can be addressed to *vertical partitioning* that uses the case dimension to partition the process over several organizations. This can be conceptualized as the “spot the difference” metaphor such that, while the commonalities among organizations are analyzed, the deviations are handled as process interleaving [15]. This setting also results in the challenge of dealing with the variability among the organizations.

2.3. Similarity Measurement and Process Configuration

As the organizations reach higher maturity levels at BPM applications, they tend to accumulate extensive number of reference process models. Actually, these models constitute a valuable asset or intellectual property to business process improvement [17, 18]. Moreover, new legislations such as Sarbanes-Oxley (SOX) emphasize on the corporate governance and operational efficiency to audit the organizations [29]. This close monitoring of processes refers to the concept *business alignment* which is related to the conformance between the apriori reference process models and the process enactments.

Process models are mostly not created from scratch and the incompliance with business requirements and the duplications of process models should be avoided. Hence the size of process model repositories enforces the automated process similarity measurement and process querying. In addition to process model refactoring by the ERP vendors, multi-national organizations can easily localize or identify more specialized processes by this process query functionality [30]. Current efforts in BPM community focus on process similarity measurement based on the task labels or the dependencies between the tasks at the reference process models [21]. While process model matching is inspired by the schema and ontology matching [19], these approaches may not sufficiently take the process behavior into account [18] such that, two processes may look quite similar in terms of task labels and the process structure, but may behave differently. Hence the analysis of behavioral similarity is complicated by two perspectives:

- There is a large variety of languages and notations for process modeling. The lack of associations among these languages results in significant discrepancies and subtle semantic issues [31].
- Task labels can be formulated in terms of different grammatical ways with syntactically different terms [32].

The classical approaches to compare the process behaviors focus on the dynamics of process models by constructing the set of process behaviors into a state space or by enumerating all possible traces. As the weakest notion, trace equivalence considers the process equivalence if the set of traces is executed in the identical way. This aspect is not feasible, since the underlying trace set should be finite, i.e. the number of traces needs to be bounded [31]. Trace equivalence also ignores the moment of choice by overemphasizing the order of activities [20]. As a relaxation for trace equivalence, bisimulation attempts to capture the inter-leavings in polynomial time [20]. Consequently, these process similarity measures aim at a true/false response rather than the degree of similarity. Moreover, they interpret all components of the process model as equally important. However, there should be a balance between rarely active and significant fragments of a process model, likewise as indicated by the process vein and process arteries analogy of De Medeiros et al. [20].

In addition to the process equivalence, *process diagnostics* is another challenging topic in process mining. According to control-flow discovery perspective, process diagnostics encompasses process performance analysis, anomaly detection, inspection of interesting patterns [21, 23]. Research at diagnosing processes is focused on finding appropriate approaches that analyze the processes in order to detect diagnostic information over some performance metrics [21]. Most real-life business processes are not strictly delimited by the underlying process-aware information systems and highly-deviated processes constitute the *flexible environments* [33]. These environments are characterized by the allowance of a wide range of process behavior, which causes a stereotypical unstructured process models called *spaghetti models*. One major factor that contributes this diversity at process execution is the *tacit process variant assumption* [34]. Therefore, the diagnostics of the processes at the model level is time-consuming and infeasible when dealing with flexible environments. Respectively, a viable solution for a better understanding of process semantics is to take care of the process semantics at the event logs and to find similar sequence of activities common across the traces. These fragments signify some sort of common functionality assessed by the process [7].

When contemporary BPM systems are evaluated in the context of cross-organizational application, the technology of BPM is lack-of-content, which means the generic solution is inadequate to support out-of-the-box business requirements or customer-specific processes [12, 14]. Likewise, ERP vendors tend to serve best practices in industry-specific adhoc applications. Such solutions are configured throughout a time-consuming and relatively complex configuration phase and these configurations make it impossible to elaborate the interdependencies among certain parameter settings. ERP vendors

tend to feature *data-centric* solutions to support particular processes at the organizational knowledge base [12, 14]. However, their large installment core and complexity make it hard to refactor and align their software components towards *process-centric* form [12, 14].

Due to the limitations of these information systems, this process-centric aspect requires a systematic treatment of *process configuration* and *configurable process modeling* [101]. Process configuration is concerned with managing the business process families that are partially or totally similar with respect to some aspects. The basic idea is to build an abstract and generic model that unifies the variances among the corresponding process family [13]. A configurable process model describes a family of similar process models and can be evaluated as the root of the underlying family. All variants in the family are derived from the configurable process model thorough a series of configuration [13]. Figure 2.4 depicts the relation between process configuration and configurable process models.

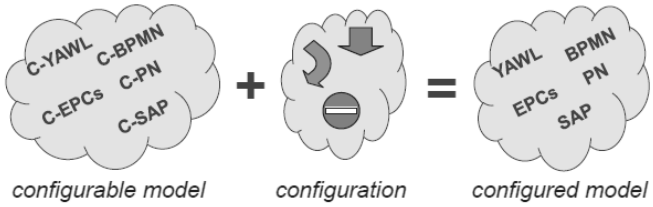


Figure 2.4. The Relation between Process Configuration and Configurable Process Models [13]. The configured model has less behavior due to the removal of potential behaviors during configuration. In other words, the desired behavior needs to be carved-out throughout the process configuration steps.

Despite the fact that; the reference models provided by ERP vendors offer little support for design by reuse, configurable process models can be built upon as the least common multiple of process variants [16]. Additionally, the analogy with the *inheritance of dynamic behavior* at object-oriented programming paradigm enriches the process configuration concept such that, each superclass of the subclass (i.e. reference process model) can be evaluated as the configured process variant as shown in Figure 2.5. That means configuration is the *inverse form of inheritance* that transforms the subclass into super-class.

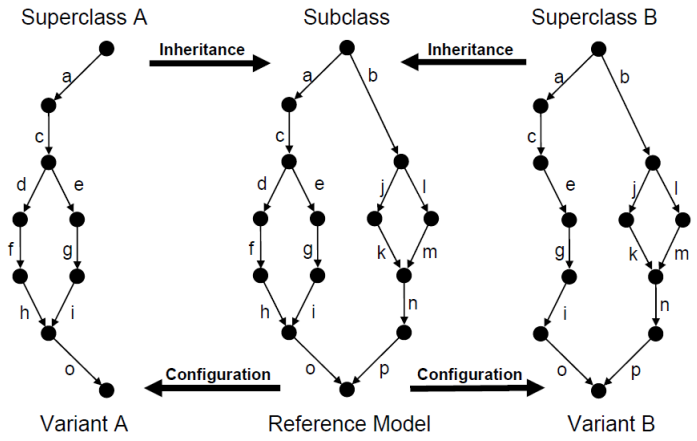


Figure 2.5. Process Configuration as the Inverse Form of Inheritance [16].

2.4. Research Questions

In order to overcome the shortcomings inherited from process equivalence notions, the purpose of this study is to design a quantitative process similarity metric to measure structural and behavioral similarities with respect to the process behaviors emphasized at the event logs. In this aspect, it is aimed to introduce a new concept named *dominant behavior*, which is a common subsequence of activities at event logs that are found to recur across the process instances, and use this runtime information to compare process alternatives.

Actually this concept respectively highlights a new perspective to the process diagnosis such that, the commonality or deviations among the process alternatives are uncovered by analyzing just dominant behavior; thereby avoiding the requirement for well-defined reference process models. In other words, dominant behavior concept enables the *log-log similarity measurement* and discovered process patterns can be compared with the patterns of other process alternatives. Hence this study widens the existing scope of process variant analysis by incorporating the actual behavior of the process alternatives, i.e. *runtime perspective*.

The region of high similarity between the dominant behaviors of distinct process alternatives might be the evidence of common functionality. [33, 34] propose the manifestation of these overlapping sub-processes as features will enable the clustering of process alternatives. Furthermore, adapting these common patterns at process configuration domain propose a way of *abstraction*, which is found a valuable feature indicated as a requirement for configurable process models [13].

In parallel with the purpose of the study, the research questions to be investigated are presented in Table 2.1.

Table 2.1. Research Questions and Details.

	Reserach Question	Details
RQ1	*What is the most common (likely) process behavior that is executed? Which proper artifact can be used for representing this process behavior?*	***Dominant behavior*** *is the most common sequence of behaviors captured at event logs that are found to occur with a process instance or across process instances with some domain significance. Respectively, compressing all of the strong-order process behaviors within a single sequence is inadeqaute. Therefore, the intra-dependencies between consecutive activities pairs, which share an incorporated business context, are also taken into account by* ***confidence values****.*
RQ2	*What is the similarity between the process models (e.g. reference and discovered process models) in a quantitative manner (0:quite difference, 1:identical)? What kind of structures can be used for process clustering?*	*Prior process equivalence notions provide an atomic answer rather than the degreeof similarity. Hence it is aimed to measure the similarity on a continous scale. Additionally, the differences between frequent and infrequent sequences are handled by the confidence concept and partial fits are taken into account. The corresponding similarity scores can be converted into* ***distance attributes*** *to be used as the baseline in segregating the process alternatives by various clustering algorithms. Alternatively, it is aimed to depict the commonalities in terms of* ***process family tree****, which is a dendrogram-like guide tree that progressively captures the similarity (and distance) among the process alternatives.*

CHAPTER 3

LITERATURE REVIEW

Process mining is an emerging discipline that provides a wide-range of approaches to discover patterns by distilling event logs, which are the baseline for end user behavior analysis [1, 4, 5, 6, 7, 8]. Process mining is anticipated as the remedy to handle discrepancies between the process enactment and reference process models created at process design phase [1]. In this chapter, process mining is handled in three categories. Prior studies about the process discovery field are given in Section 3.1. In Section 3.2, it is aimed to cover two major aspects in process similarity measurement: sequence alignment adaptations in process mining and delta analysis approaches based on the sets of traces. Finally contemporary aspects in process configuration are described in Section 3.3.

3.1. Process Discovery

Prior studies in process discovery field can be classified into three prescriptive types:

- *Correlation-based approaches* focus on extracting strong correlations, frequent patterns, associations or casual connections among activities in the event logs.
- *Classification-based approaches* aim to induce a rule set from event logs and build a classifier to predict the type of log-based relations as causal (c), exclusive (e), parallel (p) and inverse casual relation (i).
- *Clustering-based approaches* mode each event as an observation at a properly identified space of features in term weights and construct process model by combining the corresponding clusters, which hold transactions sharing the same structure and the same unexpected behavior [34, 58, 59].

The idea of applying process discovery in the context of process mining was first introduced in [4]. In this study, two key points are handled. The first point is to discover a process structure generating activities appearing in a given event log set. The second one is to define the relational conditions. As a shortcoming, there is not any requirement to identify the nature of AND, OR and XOR gateways according to the nature of process structures. Moreover process graphs are acyclic. The sole way to deal with these iterative process behaviors is to list and enumerate all initiatives of the underlying activity [17, 20, 50]. Unfortunately, this requires unifying the activity labels and activity occurrences to eliminate the redundancies.

Cook and Wolf investigated similar issues in the context of software engineering domain. In [5] they designated three approaches for process discovery ranging from the purely algorithmic to purely statistical: one using neural networks named RNET, one using a purely algorithmic approach named KTAIL and one Markovian approach named MARKOV. The approach covering these three methods is to handle underlying process discovery issue as a *grammar inference*. Respectively, the event logs

characterizing the process behavior are transformed into common sentences structures in the corresponding language and the grammar of this language is then restricted according to the formal process modeling notations. Major shortcoming of this grammar inference is that the underlying methods generally do not support seeding the underlying algorithm with aprori information about the process model in order to formulate the major process structure. Additionally grammar inference methods focus on a single state machine. In the typical process model, activities generally occur concurrently, which produce a process stream that may have non-deterministic orderings of activities. Cook and Wolf consider KTAIL and MARKOV methods as the most promising approaches, while RNET method is not mature to be used in practical applications. KTAIL method builds a finite state machine where states are complex if their successive behaviors are identical. Finite state machine (FSM) is the preferred representation in this study not to make software process prescribing more sophisticated. Actually FSMs are quite convenient, relatively simple and sufficiently powerful for describing historical patterns of actual behavior. Additionally, the results presented in [5] are limited to sequential behavior.

On the other hand, the technique of Weijters and Aalst [7] can deal with noise at the event logs and can also be used to validate business processes by uncovering and measuring the discrepancies between the prescriptive models (e.g. reference models given in business blueprints) and actual process executions. Compared to Cook and Wolf's prior approach, Weijters and Aalst focused on business processes with concurrent behavior, i.e. detecting concurrency is one of the fundamental concerns. Therefore AND/OR/XOR gateways are aimed to be explicitly distinguished in the process model. To accomplish this goal, WorkFlow nets, which are a subset of Petri nets, is integrated with techniques from machine learning. Moreover Weijters and Aalst proposed local and global metrics, which are quite distinct from the proposed metrics (i.e. entropy, event type counts, periodicity and causality) given in [5], to find explicit representations for a broad range of process models.

Proposed technique in [7] is composed of three steps: (i) construction of *dependency/frequency table*, (ii) generation of a *dependency/frequency graph* based on the dependency/frequency table and (iii) revision of the Workflow net out of dependency/frequency graph and dependency/frequency table. Dependency/frequency table composes the following attributes from event logs: the overall frequency of activity A (notation of #A), the frequency of activity A directly preceded by task B (notation of #B<A), the frequency of activity A directly followed by task B (notation of #A>B), a *local metric* that indicates the strength of the dependency relation between activity A and activity B (notation of $A\rightarrow^{L}B$) and A more *global metric* that indicates the strength of the dependency relation (notation of $A\rightarrow B$) as stated in [7].

After dependency/frequency table is constructed, dependency scores between activity pairs are calculated. As the last step, dependency scores and the information at dependency/frequency table are combined to detect the types of AND/OR/XOR gateways. Respectively, dependency score, which is approximately equal to total number of incoming or outgoing transitions of the underlying activity, implies an AND-connection, while dependency scores complementing each other to total number of incoming or outgoing transitions of the underlying activity implies an OR-connection. Unfortunately proposed mining technique in [35] has still limitations with handling complex interleaving process structures in combination with short (one-step) loops. Proposed mining technique in the underlying study is realized as a tool named Little Thumb.

Formal approaches stated above are based on the assumption of a weak notion of completeness and noise-free event log. Actually in practical settings, event logs are rarely complete and noise free. Hence HeuristicMiner approach stated in [36] anticipates three threshold settings to handle this issue: (i) the dependency score, (ii) the positive observation score and (iii) the relative to best threshold. Approaches that lack of the capability of detecting the nature of AND/OR/XOR gateways suppose that the corresponding threshold setting is unnecessary for dependency relations according to "all activities connected" heuristic. As the major novelty of [36], Weijters et al. proposed a solid measurement to express the type of splits and joins instead of the intuitive heuristic approaches like in [35].

In the following study [2], Weijters and Aalst introduced two additional parameters: noise factor *N* and a threshold value δ. The value of δ is derived from N, the number of lines in the event logs (#L) and the number of activity types in the related business process. Then frequencies given in dependency/frequency table are compared with δ to determine whether underlying relation is adequate to be indicated in the process model. Weijters and Aalst enhanced a novel approach to the present approach in [7], which resides in the fact that they use a global learning approach, named logistic regression model and find a threshold value that can be used to detect direct successions in [3]. As the basic material, dependency/frequency table is used as in [7]. Addition to existing parameters in

dependency/frequency table, the frequency of task B directly succeeded by another task A, but before the next appearance of B (i.e. notation of B>>>A) and the frequency of task A directly succeeded by another task B, but before the next appearance of A (notation of A>>>B) parameters are taken into consideration.

Additionally local and global metrics introduced in [7], which indicate the local and global strength of the relation are revised in [3] and a new metric, namely causality, is introduced, n is the number of activities between A and B, then causality metric is incremented with a factor δ^n, where δ is a causality factor, i.e. δЄ[0,1]. Respectively, causality is an adaptation of global metric introduced in [7] in terms of long term successions.

Model emphasized in [3] is to harmonize these three metrics described above and to find a threshold probability P to determine whether two activities A and B can are in type of direct succession or succession relation. Global learning method proposed in [3] uses information embedded in event logs to interpret the direct successor relations between events. This method is able to find almost all direct connections in the presence of parallelism, noise and an incomplete event log. In [24] Maruster, Weijters and Bosch implemented another adaptation of this method on simulated hospital event logs, containing information about which medical actions took place over time. Technique in [24] cannot cover all kind of Workflow nets, as shown in one experiment involving none-free-choice showed.

In [6], the goal of proposed method, named *alpha algorithm*, is twofold: first of all, a mining algorithm is sought to rediscover sound Workflow nets, i.e. based on a complete event log the corresponding workflow process model can be derived without any extra behaviors. Second, given such an algorithm, it is aimed to detect the type of the rediscovered workflow nets. Clearly, this class set should be as large as possible. Note that in the prior studies [2, 3, 5, 7, 37] there is not any mining algorithm which is able to rediscover all sound Workflow nets. As a way of representation, Maruster, Weijters and Aalst attempted to construct concrete Petri net for a broad range of process models rather than a set of dependency relation between events like in [7].

Actually the preliminary results presented in [2, 3, 7, 37] only provide heuristics and basically handle on issues such as noise, basic parallelism, basic closed loops. The approach described in [6] differs from prior approaches in the sense that; it is proven that for certain subclasses (e.g. non-free choice, basic and arbitrary loops, hidden tasks, noise, basic and complex parallelism) it is possible to find the right process model by alpha algorithm. Also this algorithm can mine timed event logs and interpret several kinds of temporal information (e.g. waiting/synchronization times, flow times, utilization) to performance metrics. On the other hand, the major limitation of alpha algorithm is that certain kind of similar tasks having the same title cannot be detected. In other words, task labels are not unified.

In [10], distinct tools, which are driven by different problem areas in process mining, are described as follows:

- EMiT (Enhanced Mining Tool) is a graphic-based process model tool that includes various type of performance metrics. Due to its graphical-based structure, it is able to handle rediscovery problem effectively.
- Little Thumb, which is firstly introduced in [7], concentrates on incomplete event logs and noise. However at a noisy and incomplete situation, single erroneous events can completely deteriorate the derivation of a right conclusion. For this reason Little Thumb is a heuristic-type mining technique which is robust to noise and the incompleteness issues at the event log.
- Although approaches previously presented assume that each task should be labeled with a unique task identifier within the process in the graphical models, it is not possible to assign multiple blocks addressed to the same task. InWoLvE (Inductive Workflow Learning via Examples) attempts to deal with duplicate tasks with lattice of task mappings in the event logs, which is inherited from machine learning and grammatical inference. Between the mappings, there is a partial ordering and the corresponding mapping lattice is featured by only the most or the least general specific likelihood element.
- Process Miner, exploits the properties of block-structured workflows with a composition of nested blocks. These blocks are characterized by the operands and constants. While constants refer to the tasks or sub-processes embodied at the underlying process, operands determine the process flow or process behavioral characteristics.

In [38], Aalst et al. aimed to validate the applicability of process mining in other practical areas. The industrial application in this study involves one of the twelve offices of the Dutch National Public Works

Department, which is primarily responsible for the construction and maintenance of the road and water infrastructure in its providence. The focus of this study is not limited to the control-flow perspective. In this case study, the organizational and case perspectives are also handled. As a supporting tool, ProM framework, which integrates EMiT, Little Thumb and MiSoN tools, is introduced in this application.

As a hybrid methodology in process mining, Gomez et al. introduced Application Usage Mining concept in [26]. Application usage mining is explained as the analysis of the user's behavior in the business application systems (e.g. ERP) by applying the basic approaches, notions and methods dedicated to web usage mining. The major distinction between web usage mining and process mining is such that, while visitors have the freedom to navigate through the web pages, the employee should optimally perform the assigned tasks along with the usage of the business application systems.

Alternatively, Weijters and Aalst introduced a new progress of process discovery issue at a more robust and confident level in [3] using a data-driven approach called *logistic regression model*, which is capable of tagging the causality relations at the event logs. The corresponding logistic regression approach requires an input threshold value that is used to determine whether there is a direct regression among the tasks. The usage of global threshold emerges some shortcomings about the robustness problem. In this aspect, [24] aims to use machine learning techniques to perform classification rules for (i) casual relations and (ii) parallel/exclusive relations assuming the existence of noisy information in event log and imbalance in execution priorities. The instantiation of a so-called dependency/frequency table from the event log information is the starting point of the method likewise in [7]. Afterwards three relational metrics, i.e. causality metric (CM), local metric (LM) and global metric (GM), are calculated for each activity pair occurred in process instances. Relational metrics and dependency/frequency table materials based on prior studies in [7] and [38].

Actually the causality, local and global metrics have been developed specifically to be used as predictor attributes for determining decision rule sets. They are less practical predicates for deciding the type of process behavior types. Last operation in [24] is to detect the existing log-based relations between tasks by applying the predictive features of the introduced metrics to the learning schema generated in dependency/frequency table. In this operation Ripper is chosen as the appropriate learning algorithm, which induces minimal description-length rule sets.

Because of supervised nature of classification, a training dataset has to be provided, each of which has been labeled with a target class. Each instance in training dataset is labeled according to the log-based relations that can exist between two tasks: (c) for causal, (e) for exclusive, (p) for parallel and (i) for an inverse casual relation. As a result, the contribution of [24] is to complement the work reported in [6] such that, it resolves shortcomings of the alpha algorithm, in dealing with issues about causality and parallel/exclusive relations exhibition in noise and incomplete process logs.

Respectively, correlation and classification based techniques disregard the non-structural event log data that is still kept by various information systems. The corresponding data composes of information about activity executors, timestamps, parameter values, as well as different performance measures.

In [39] Chiaravalloti et al. presented an enhanced process mining approach, where different process alternatives are discovered by segregating the process streams according to structural attributes and performance metrics. These behavioral and performance measures are presented by proper auxiliary domains. The basic issue about this multi-auxiliary domain is the quantifying the relevance of these domains.

In parallel to [39], [40] concentrated on the adaptation of data mining techniques for process mining through hierarchical clustering of the event logs, in which each trace is featured as an observation of a properly identified space of features. As a major distinction in this study, previous approach in [39] is extended by proposing a process discovery algorithm that both discovers the behavioral structural of a given business process and enrich the discovered schema with some interesting global constraints. These constraints are relatively rich in notation and highly correlated to the corresponding business context in process structure. Thus these global constraints are often expressed using other complex formalisms, mainly associated with clear semantics.

Lastly, [41] aimed to precisely investigate the unconnected process patterns, which are sets of the behaviors that frequently occur together in some event log data. The corresponding approach uses a set of frequent patterns as input and discovers the interconnections at the subset of these frequent pattern set. Proposed technique in [41] can be used for unifying sets of arbitrary sub-processes that

are very often executed together and may be abstractly focused as a sub-process in the workflow schema. Hence these unconnected patterns can be used to denote interesting and useful correlations among sub-processes which are seemingly not related with each other.

In [42], a new approach for process discovery was introduced by adapting from-to chart for analyzing the event logs. This approach is composed of two components: from-to chart and process flow branch discovery. From-to chart is an analytical tool, which is basically used in monitoring material handling routes between operations, machines, departments or work centers on the production floor. The underlying approach inherits this tool from facility layout domain and adapts it in process discovery field as the basic bookkeeping material in monitoring transitions among activities occurred in process instances and figuring out if there exists any specific order of the occurrences for representing in process model. In [43], underlying process discovery approach was further improved by Genetic Algorithms for rearranging the from-to chart in order to search the sub-optimal arrangement at process modeling. In the previous work, this rearrangement operation is performed by a permutative fashion, which leads to an exponential increase in total processing time at handling relatively complex business processes. In [44], Esgin and Karagoz extended the work in [43] by extraction of AND/OR/XOR gateways. Through this extension, the type of connections are discovered for each parallel predecessor or successor of underlying activities by interpreting the structure of dependency/frequency graph and the final scores at from-to chart. Hence dependency/frequency graph is converted into a block-oriented model named control flow graph.

3.2. Process Similarity Measurement

As the starting point, we focus on the surveys at process model similarity domain [19, 32]. In [19], Becker and Laue presented a comparative study about business process similarity and concentrate on four major trends in this domain: (i) approaches based on correspondence between process model elements (i.e. nodes and edges) (ii) the applicability of graph algorithms on similarity calculation (iii) causal dependencies between activities in process models and (iv) approaches based on the sets of traces. The survey given in [19] served as a complementary paper and commentary to [30], which is one of the first major conference papers on this topic. Dijkman et al. [19] highlighted the challenge of process model similarity even if process models depict exactly the same behavior at the same process detail level with the same objective, this similarity measure might be a combination of behavioral representation, task labeling styles and modeling notations.

In [30], Dongen et al. defined the concept of *causal footprint*, which is a set of essential behavioral restrictions determined by the process structure. The causal footprint combines two distinct relations among the nodes: *look-back links* (i.e. the execution of the source of underlying link leads to the execution of at least one of the targets) and *look-ahead links* (i.e. the execution of the target of underlying link is preceded by the execution of at least one of the sources). For calculating similarity, corresponding vector model, which is a major technique in information retrieval (IR), is adapted. In this adaptation, the set of terms is built upon the union over nodes, look-back and look-ahead links and the weights are determined due to the size of the terms. The similarity between footprint vectors is measured by the *cosine similarity*.

In [31], Mendling et al. extended this abstract process representation of the process behavior with causality graphs and causal closure concepts. Rather than verifying the entire process model, causality graph captures the approximate intended behavior of the process at a high level. Causal closure holds the smallest possible span of this causality graph. Mendling et al. emphasized the advantage of causal footprints such as; these process abstractions hold the information about the sequence of activities according to their direct succession. Additionally they are robust with respect to the problems such as termination or finite size issues of state space which determines atomic behavioral similarity measurements. In [45], Dijkman et al. made extensions to causality footprint technique by introducing two additional similarity metrics: *label* and *structural similarity*. While label similarity is obtained by calculating the optimal equivalence mapping between the nodes of the process models being compared, structural similarity is based on *graph edit distance*. This technique searches the minimum number of edit operation (i.e. node deletion or insertion, node substitution and edge deletion or insertion) that convert the given process structure to the target one. In [17], Dumas et al. reviewed the NP-hard computing of graph edit distance in structural similarity by *A* algorithm*, *heuristics search* and *similarity flooding*. A* algorithm progressively builds up the partial mapping of larger size graph, until the instance of a larger mapping happens infeasible with a lower edit distance. In this basic form, this algorithm constructs one-to-one node mappings by considering elementary edit operations. Respectively, heuristic search is a greedy technique that iteratively maintains a mapping list holding the most similar nodes without any existence at current mapping state. Alternatively, similarity flooding holds the pair of nodes and edges with their adjacent neighboring elements.

In [20], Medeiros et al. presented major disadvantages of prior techniques in process similarity measurement, i.e. trace equivalence and bisimulation. In trace equivalence, two process models are considered equivalent in the case of identical execution logs. This notion is seemingly erroneous in two aspects: (i) the set of traces may be infinite (ii) trace equivalence cannot catch the moment of choice. Bisimulation can be performed in polynomial time. In addition to these shortcomings, these equivalence notions result in binary answer instead of the degree of similarity. Hence they introduce the concept of *observed behavior*, which enables to compare infinite number of execution sequences and consider the relevance among these traces. This concept is in parallel with the *behavioral pseudometric* to evaluate the transition systems as shown in [46]. Since observed behavior checks the enabled transitions (i.e. the moment of choice) in process models, this idea results in the *behavioral precision and recall* metrics.

Aalst proposed an abstraction for the matching between observed and modeled behavior named frequency profile in [47]. Indeed process execution may deviate from the prescriptive or descriptive process model given at business blueprint. Additionally at various contemporary information systems, there is a lack of process monitoring functionality to integrate the transaction logs with corresponding use-case. Aalst formulated an integer programming (IP) to check whether the modeled behavior and the observed behavior match. IP is built upon Petri-net firing rule and frequency profiles as the

constraints. Likewise in [47], Kunze et al. proposed a behavioral profile, which is a Jaccard coefficient related metric to measure process similarity in [48]. This metric evaluates the behavioral relations between pairs of process activities in the context of strict order, exclusiveness and interleaving order relation.

Current tendencies in BPM field covers the similarity measurements based on the semantic and syntactic correlations between the task labels and behavioral features obtained from the control flow aspect of the corresponding business process. This means an adaptation of the algorithms from the fields of information retrieval and graph theory. On the other hand, approaches based on execution traces or event logs (e.g. the approaches emphasized in [20, 47]) may not adequately take the behavior of a process model into account, since process executions cannot be logged through a case perspective as stated in [47]. This rationale is called *unlabeled event* logs. In [18], Wang et al. defined incidence matrix and coverability tree concepts built upon Petri-nets and introduce the concept of principal transition sequences (PTS) to construct a good conceptualization for the essence of the behavior of a process model. While coverability tree is one of the fundamental methods for behavioral analysis of Petri nets to overcome infinite state space and reachability tree dilemma, principal transition sequences act as a characterization of transition sequences that lead from the initial marking to the final state. The similarity of principal transition sequences is based on longest common sub-sequence concept.

In parallel to [18], Zha et al. emphasized the concept of transition adjacency relations (TAR), which are the genes of firing sequences in Petri-nets [49]. Although firing sequences are a good process approximation for the behavior of process behavior, they can be impractical to explore the high complex state space of concurrent process models. Therefore prior approaches such as causal footprints [30, 31, 45] or observed behavior [20, 47] are based on substitute representations. However the results of these similarity notions are incomparable, because there is lack of consensus on the process similarity concept. In this aspect, TAR is not like a look-ahead and look-back relation given in [30], which can be derived from the process structure. The generation of TAR set requires exploring the state space of a process. TAR can be instantiated by reachable marking graph of a given Workflow-net. At this graph, each node represents reachable marking states and each edge implies a process flow. On the other hand, instantiating this reachable graph is computationally expensive, since the complexity of this graph is exponentially correlated with the number of nodes.

Measuring compliance with reference process models is also a significant issue in process similarity measurement. Gerke et al. highlighted the limitation of existing approaches for measuring compliance as the assumption that the compliance is solely based upon the notion of process equivalence in [50]. This is due to (i) difference at the process granularity, (ii) partial or limited view of process mining and (iii) overemphasis of the order of activities. In this aspect, Gerke et al. defined process compliance in the context of compliance degree and compliance maturity. According to Rosemann [51], finding the exact level of detail in process modeling is one of the potential pitfalls in BPM community. Vanderfeesten et al. [52] elaborated on quality metrics for business process modeling and emphasize the adaptation of software engineering related metrics (i.e. coupling, cohesion, complexity, modularity and size) to process similarity measurement. We propose a set of structural influence factors, which are based on the metrics given in [50, 51, 52], to analyze the understandability of the process models.

In [29], Rozinat and Aalst tackled the conformance checking problem between descriptive (or prescriptive) process models and process execution in two dimensions: *fitness* and *appropriateness*. While fitness measures the association between the event logs and process execution variants, appropriateness is the degree of simplicity in which the process model describes the observed behavior. Rozinat and Aalst revisited the quality of process discovery in [53]. They propose new quality perspectives (e.g. accuracy, process minimalism and completeness) and noise generation based on Hidden Markov Models (HMM).

Although the goal of process mining is to discover process model, these models tend to be very confusing and difficult to understand for relatively flexible environments. These generated models are usually called spaghetti models. In this aspect, trace alignment aims to align traces (i.e. finite sequence of activities) in such a way that the event logs can be analyzed easily. This operation can be designed as a preprocessing phase where the event logs are interpreted, filtered or divided into distinct clusters. Hence it complements current process mining approaches that focus on process discovery and conformance checking functionalities.

In [23], Bose and Aalst presented initial success stories demonstrating that emerging process mining discipline can benefit from techniques developed for Bioinformatics. They adopted the sequence patterns (e.g. sequence motif such as tandem repeats and maximal repeats) in bioinformatics and proposed a means to form abstractions over these patterns. Using these abstractions as a basis, a two-phase approach to process discovery is introduced. The first phase preprocesses the event logs according to the process abstractions obtained for a predefined detailed level and the second phase aims to discover the process model with an adaptive zooming functionality. As a result, trace alignment can assists in answering a variety of process diagnostics questions.

Bose and Aalst handled the *process diagnostics*, which is one of the challenging topics in process mining in [21]. Process diagnostics covers a wide range of applications such as process performance analysis, anomaly detection, diagnostics and inspection of interesting end-user behavior patterns. When dealing with concept drift in real-life processes, diagnostics of processes at model level may turn out to be infeasible and time-consuming. Trace alignment approach that is introduced in [23] is discussed towards pairwise and multi-trace alignment aspects. While biological sequences tend to be homogeneous, heterogeneity of event logs and variation in the length of these event logs for semi-structured environments (e.g. health care industry) may result in impractical dynamic programming implementation for multi-trace alignment. Therefore Bose and Aalst adopted *sum-of-pairs* (SP), which is one of the most popular scoring mechanisms for multi-sequence alignment of genomic sequences.

As stated before, traditional process mining algorithms have various shortcomings in coping with complex spaghetti-like process structures, which are hard to interpret and visualize. In [21], a context-aware approach is proposed to overcome this problem. The approach, namely *generic edit distance framework*, aims to segregate process traces in such a way that; each trace cluster builds up a lean *lasagna-like* process model. Additionally several approaches for trace clustering, i.e. bag-of-activities, k-gram model and hamming distance, their issues and challenges are investigated in [21]. While prior approaches do not consider the functional validity of any edit operations, generic edit distance framework proposes a robust cost function that avoids edit operations that are infeasible in the business context. Bose and Aalst proposed two quality metrics to evaluate the goodness of trace clustering: (i) generated process models should have a high degree of fitness (ii) the process models should be less complex.

In [54], Stolfa et al. implemented sequence alignment methods to a real-life use-case, namely SAP invoice process. They aimed to adjust sequence alignment to be able to determine similarity between distinct business processes. In this aspect, they compared the longest common substring (LCS), the longest common subsequence (LCSS) and the time-wrapped longest common subsequence (TWLCS). While LCSS is more tolerant to slight distortion in the sequence ordering than LCS, TWLCS is more robust to minor distortions and to time non-linearity. Additionally they performed a quite distinct data preparation procedure where the events are categorized according to their duration and activity types (e.g. verification, creation, approval and posting).

Likewise in [54], Goa et al. emphasized the NP-hard computational complexity of LCS in measuring the similarity of traces in [55]. Hence they extend the Hungarian method to select the best matching that maximizes the sum of semantic similarity degrees between activity pairs. Then approximate longest common sub-trace is defined to measure the commonality of traces. Juan applied string coding and comparison to analyze the process logic difference between business processes in [56]. Process paths embedded into process models are identified and encoded into process path strings. Process path strings, which are filtered by semantic similarity degree (SSD) threshold, are analyzed in three concerns: unique activities, processing mode and processing sequence.

In [34], Song et al. demonstrated that proposed trace clustering approach, based on *event log profiles*, i.e. activity and originator profiles, can improve process mining results in real flexible environments. The proposed *divide-and-conquer approach* is based on a set of profiles, each quantifying a number of features from a specific perspective. Based on derived feature matrices, several distance metrics (e.g. Euclidean, Hamming and Jaccard distance) are applied to compute relative distance among use-cases in the event log. Quality threshold clustering, which determines the maximum cluster diameter with respect to quality threshold, and agglomerative hierarchical clustering (AHC), which is usually illustrated by dendrogram, is implemented at trace clustering step. This profiling paradigm is also handled in [57]. Rao et al. introduced a profile for a multi-sequence alignment as a sequence of compositions and each composition holds the frequencies of each character (activities) at alignment. Indeed, the relative distance of distance functions with respect to alignments reflects a distinct aspect in evaluating distance function to cluster tandem repeats.

In addition to generic edit distance metric introduced in [21], Bose and Aalst described the concept of conserved patterns, which are sub-sequence of high similarity shared within process instance in [58]. These regions are formalized by various features, i.e. maximal pair, maximal repeat, super maximal repeat and near super maximal repeat. The authors also suggested several statistical metrics in evaluating the significance of clusters such as average cluster density and silhouette width. Primitive tandem arrays and conserved pattern features introduced in [58] are converted into equivalence classes in [33]. Then abstractions of conserved patterns are depicted as Hasse diagram. Due to dealing with complex constructs, the exact conserved pattern definitions are relaxed through approximate definitions and efficient suffix-tree constructions are adopted to handle very long event sequences.

As stated in [21, 58], multiple sequence alignment of genomic sequences, namely sum-of-pairs methods, is applicable to improve the NP-complete problem. But current trace clustering methods suffer from the divergence between clustering and evaluation biases. In [59], De Weerdt et al. addressed this gap by an active learning approach that concurrently mines and evaluates process models during clustering step. Hence this concurrent and proactive trace clustering is accomplished by a forward-looking procedure that only adds process instances with better fitness score to current cluster. Instances that are not assigned to current cluster are handled at the following selection and look-ahead iteration. Alternatively, sequence mining method introduced in [60] proposes to cluster traces by a learning model that combines first-order Markov models with expectation-maximization (EM) algorithm. Hence the assignment of sequence to clusters is determined according to the probability of each cluster to generate the given trace. Additionally in [60], Veiga and Ferreira revised the preprocessing steps by discarding the most recurring events and sequences. This is motivated by the fact that spaghetti models are chaotic due to the contribution of these frequent events.

Respectively, both methods in [59, 60] are computationally expensive. Hence to handle the bias between clustering and evaluation, Evermann et al. focused on designation of a distance metric that allows the adaptation of generic multivariate clustering method in [61]. This proposed approach, i.e. AlignCluster, uses the Smith-Waterman-Gotoh algorithm for sequence alignment to compute process similarity, applies multi-dimensional normalization to construct a feature set for vector representation and applies K-means clustering in oppose to agglomerative clustering applied in [34, 57, 62]. Underlying trace clustering method removes duplicated traces and assigns equal weight to each unique trace at preprocessing step. The authors evaluated four quality dimensions of fitness, precision, generalization and simplicity. Simplicity dimension is enriched with three new metrics: the cyclomatic number, the coefficient of connectivity and the density.

Actually dropping or pruning the events and sequences with low support emphasized in [60, 61] is in parallel to frequent and strong sequence concept in [63]. Lesh et al. adapted sequence mining as a preprocessor to determine feature set for standard classification algorithms such as Naïve Bayes and Winnow. In [40], Greco et al. introduced a process mining framework to identify different variants of the underlying process. It is an iterative, hierarchical refinement of process discovery, where traces with similar behavior are clustered together with a specialized schema called workflow schema (WS). The quality of mined model is evaluated according to two quality metrics: *completeness* and *soundness.* Definitely, a complete process model is such that all event logs are compliant with some instance of the model (similar to *fitness*), while a sound model implies that all possible enactments are registered by the event logs (similar to *minimality* or *behavioral appropriateness* given in [24]).

In [64], Mendling and Strembeck discussed the process understandability as a particular quality aspect in twofold manner. First, three factors categories (i.e. personal, structural and textual) are identified in order to evaluate understandability issue. According to experimental findings, process observer's background (*theory* attribute) and the underlying process model's separability feature (*separability* attribute) are positively correlated with process understandability. On the other hand, activity label length (*textlength* attribute) has a negative significance. In this study, it is also aimed to analyze the correlation between the professional experience of process observers and the similarity measurement concern of proposed approach. The analysis, which interprets the correlation between the professional experience of process observers and the similarity measurement concern of Pairwise Alignment, is based on an analogy with theory attribute given in [64].

Esgin and Karagoz [65] proposed a distance metric, which is built on the vector model from information retrieval and an abstraction of process behavior as *process triple*. Process triple is a set that covers transaction existence and interactions (successor/predecessors of each transaction) among activities. This metric takes structural and behavioral perspectives into account. Alternatively in [66], it is aimed to demonstrate that process similarity measurement can benefit from sequence mining techniques,

which are strengthened with standard Needleman-Wunsch algorithm to quantify the similarities and discrepancies. Unlike [65], the proposed approach evaluates just consensus activity sequences by avoiding the requirement for well-structured process models. A new alignment approach called Confidence Aware Needleman-Wunsch (CANW) is introduced in [67] by the determination of insertion/deletion (inDel) scores in business context-aware fashion according to the interactions among activities. In [68], match/mismatch scoring is revised in such a way that opportunity cost function is introduced for replacement of current prefixes that are quite different. Consequently, the adaptation of sequence alignment to process mining domain has highlighted a new perspective to similarity measurement; deviations and violations are uncovered by analyzing just event logs and thereby avoiding the requirement for well-defined reference process models.

Sequence alignment adaptation in prior studies is realized by preprocessing event logs with abstractions at a desired level of granularity. Hence event logs are split into homogenous subsets and more structured process models are discovered for each subset. Alternatively in this study, proposed framework measures the degree of process similarity on log-log sequence alignment basis and segregates the process alternatives into more homogeneous process families. Relatively significant common patterns in these facets are visualized and interpreted by new feature sets.

3.3. Process Configuration

Traditional process mining techniques focus on monolithic processes in a particular organization. However, with the emergence of new shared economical models and information systems architectures (e.g. shared business process management infrastructures (SBPMI) and cloud computing), the perspective of process mining research area is extended towards cross-organizational applications. Cross-organizational process mining is an emerging concept, due to the distribution or replication of similar processes over multiple organizations. In [15], van der Aalst explored the possibilities of intra- and inter-organizational process mining. For this purpose, two basic settings are handled: (i) collaboration and (ii) exploiting commonality. While collaboration aims to distribute the work associated to a case over different organizations throughout an interoperability manner, exploiting the commonality focuses on sharing best practices, knowledge or a common infrastructure. Additionally, these two settings are extended with *horizontal* and *vertical partitioning* concepts. Vertical partitioning is related to the case dimension to group the work, i.e. the cases are distributed over several organizations, horizontal partitioning is based on the process dimension, i.e. the process is divided into sub-processes and assigned to distinct organizations. Although these partitioning strategies are crucial for intra- and inter-organizational process mining, there emerge new challenges such as a myopic attitude at horizontal partition and the effect of infrequent (surprise-typed) events at analyzing the commonalities.

In the study of Buijs et al. [11], process models and intended behaviors of the organizations are cross-compared as a means to supplement the representation. While the capabilities of Shared Business Process Management Infrastructure (SBPMI) at cross-organizational process mining are emphasized, they introduce *dotted chart* as a mean of visualization of the process enactments of distinct organizations. In a dotted chart, each dot refers to a single event execution where the color indicates the activity type. Each row stands for a process instance; the horizontal axis represents the time. In [27], prior process variant management is extended with process alignment matrix that allows for log-model comparison which is strengthened with the feedbacks from process observers. Hence log-log comparison paradigm and incorporating the actual behavior of process variants, i.e. runtime perspective, given in [11] are enriched towards this comparison aspect. Alternatively, Yilmaz and Karagoz proposed a four staged solution in cross-organizational process mining in [28]. This framework compares groups of process variants in order to provide critical feedbacks on the potentially significant parts of the process maps. Random initialization based K-Means++ approach is used as the clustering algorithm to group the organizations. Then sum-of-squared errors are plotted as the recommendation to the process observers to determine the appropriate cluster size.

In [16], Gottschalk et al. discussed the theoretical representation for configurable process models and the dependency among these models within the context of the inheritance of process behavior. Despite the fact that; SAP (or other ERP vendors) reference models offer little support for design by reuse, configurable process models can be built upon as the least common multiple of process variants. Additionally analogy with the *inheritance of dynamic behavior* at object-oriented programming paradigm enriches the process configuration concept. Hence each superclass of the sub-class (i.e. reference process model) can be evaluated as the configured process variant. In other words, process configuration is the inverse form of inheritance that transforms the subclass into super-class.

In [12], cross-organizational process mining for configurable services in shared architectures is elaborated. Van der Aalst highlights the challenges of contemporary BPM and ERP systems to deal with the variability across organizations. While current BPM tools aim to create generic process modeling services to process-aware information systems and are not capable to response unstandardized out-of-box requirements, complex installment baseline of ERP systems are too static to be adapted from data-centric solutions towards process-centric ones [13]. As in [28], clustering is evaluated as appropriate technique to group process variants in [12]. Similarly, the dilemma between process configuration and mining is formularized in [13]. In classical system implementation, organizations make adhoc customizations to compensate their requirements. But this is undesirable for supporting cross-organizational (multi-tenant) processes. As the theoretical basis, causal nets are adapted as a new formalism to deal with the challenges in process configuration. In [14], van der Aalst et al. defined a configuration guideline (or a roadmap) to characterize all correct process configurations at design phase without any restrictions on the modeling class. Adriansyah et al. proposed a technique to allocate a penalty cost to particular deviations and find the alignment between observed behavior (i.e. event logs) and modeled behavior due to this costing in [69]. In this context, skipped and inserted activity concepts are manifested.

As a sub-issue in process configuration, Nguyen et al. presented a comparison of sequence mining techniques for *deviance mining* in [70]. This is a family of techniques to explain the reasons why underlying process deviates from proposed or expected execution patterns. The paper compares *frequent pattern mining* and *discriminative mining*. Frequent pattern mining manages the frequent structures (e.g. tandem repeat, maximal repeat etc.) as boolean features, the features in discriminative mining are traced within and across the traces. In oppose to model delta analysis perspective in [70], van Beest et al. handled deviance mining application via a *log delta analysis* perspective in [9]. Respectively, model delta analysis is based on manual comparison with discovered process models. Hence it is error-prone and inapplicable to complex processes. The method encodes event logs as event structures enhanced with frequency information.

CHAPTER 4

BACKGROUND

4.1. From-to Chart Adaptation

4.1.1. From-to Chart as a Basic Analytical Tool

As in Facility Layout Problem (FLP) domain, the basic *from-to chart* is a square matrix for summarizing material handling between related operations, machines, departments or work centers on the production floor with high volume production rate [71, 72]. The sequence of operations is written down the left-hand side of the form and across the top. While the horizontal sequence of activities is the from side of the matrix, the vertical sequence of activities is the to matrix [73]. This analytical technique is useful for designing relative locations of operations, demonstrating the material flow patterns, showing the degree of self-sufficiency of each operation, Interpreting possible production control problems, planning the inter-relationships between several products, representing the quantitative relationships between the operations, evaluating the alternative flow patterns and improving the distances traveled during a process [71].

The number of rows and columns in the matrix is equal to the number of operations under consideration. Additionally the operation titles are listed in identical order across the top of the columns and down the row on the left hand side of the matrix. Initial row or column sequence may represent geographical arrangement in the plant, logical arrangement of process flow or proposed sequence as represented in Figure 4.1.

to / *from*	Rough Stores	Mill	Lathe	Drill	Bore	Grind	Press	**Total**
Rough Stores		18	72			9	36	**135**
Mill			9	18			9	**36**
Lathe		18		36			9	**63**
Drill		9			9		18	**36**
Bore				9				**9**
Grind				9				**9**
Press				18				**18**
Total	**0**	**45**	**81**	**90**	**9**	**9**	**72**	**306**

Figure 4.1. From-to Chart as a Basic Analytical Tool at Plant Layout [71].

Basic data for entry into from-to chart are prepared by tabulating the flow paths of each part, product or material in such a way that, for each move of related entity from operation *i* to operation *j*, current score at the $(i,j)^{th}$ element of matrix is incremented by one. Thus accumulated scores at each element represent the total number of moves from and to the underlying operation. Data entry into the matrix can be calculated in several ways, depending on objective or desired result of the analysis [71]. Scores may also represent the number of moves between operations, the quantity of material moved per time period, the weight of material moved per time period, the combination of quantity × weight per time

period. Constructed from-to chart has to be analyzed for better arrangements of operations to reduce handling, costs, distances and production control problems, etc. [71]. Major use cases occurred at from-to chart are as follows:

i. All entries below the diagonal indicate *backtracking*, i.e., backwards from the order indicated by the numbers representing the operations.
ii. All entries in the upper right or far right indicate *skipping* past several adjacent operations to get to their next operation.
iii. Items moving from one operation to an adjacent operation result in the marks falling in the elements along and just above the diagonal. This represents *straight-line* (*direct*) flow.

These use cases at the from-to chart given in Figure 4.1 are visualized according to the type of use-cases as shown in Figure 4.2.

to / from	Rough Stores	Mill	Lathe	Drill	Bore	Grind	Press	Total
Rough Stores		18	72			9	36	**135**
Mill			9	18			9	**36**
Lathe		18		36			9	**63**
Drill		9			9		18	**36**
Bore				9				**9**
Grind				9				**9**
Press				18				**18**
Total	**0**	**45**	**81**	**90**	**9**	**9**	**72**	**306**

straight-line (direct) skipping backtracking

Figure 4.2. Major Use Cases Occurred at From-to Chart.

Intuitively it is seen that the best layout can be devised by rearranging the columns and rows to put the elements with relatively larger scores just above the diagonal and fewer ones below the line [71]. Indeed, this arrangement may be possible for one material, but it is not possible for all materials at production portfolio systematically.

According to [72], the from-to chart is a descriptive material to reduce a large volume data into a workable formation and the construction of from-to chart does not result directly in the solution of a layout problem. On the other hand, a more quantitative approach to minimize material handling is obtained by taking *moments* of the accumulated score at each element around the diagonal and aiming for the lowest total moment (*Z*) at the current state of from-to chart [71]. The number of elements away from the diagonal is used as the distance from the diagonal, i.e. *moment arm*[1]. Objective function to minimize the total moment of from-to chart is formulated as given in Equation 4.1:

$$Min\ Z = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij} \times |j-i| \times p \qquad (4.1)$$

While f_{ij} indicates total move (transition) from operation i to operation j, p is the backtracking penalty point assigned to each entry below the diagonal. Back-tracking penalty point is parameterized to enforce the model towards a straight-line arrangement [73].

4.1.2. Rule Induction at From-to Chart

In the traditional use of from-to chart, total score of each element is directly taken into consideration in rearrangement of the matrix. However, this state is exaggerated by the amounts of data stream being

[1] To make moment computation simple, suppose all operations (machines) are of the same size and the distance between the working points of each pair of adjacent operation (machine) is just one unit.

collected and stored at the matrix. Therefore a requirement of rule induction procedure is emerged [74]. This evaluation step aims to prune down the weak scores before rearrangement and eliminate their effect on the dominant behavior as stated in [42].

Basically, there are three evaluation metrics introduced in [42]: *confidence for from-to chart* (confFTC), *support for from-to chart* (suppFTC) and *modified lift* (ML). These metrics act as the major threshold to control the level of robustness and complexity of the discovered process model constructed from large amounts of data.

***Definition* (confidence for from-to chart, *confFTC*)**. It is the ratio of transitions that are from predecessor *A* to successor *B* (|A > B|), to the total number of transitions which are initiated by activity *A* (i.e. |A >*|, row total of activity A at from-to chart), as given in Equation 4.2. This metric is the basis for inDel (insertion/deletion) scoring at sequence alignment phase.

$$confFTC(A,B) = \frac{|A > B|}{|A > *|} \tag{4.2}$$

confFTC metric is similar to the *local metric* (LM) introduced in [3]. LM implies the probability of succession relation by comparing the value of |A > B| versus |B > A| as given in Equation 4.3.

$$LM = P - 1.96 \times \sqrt{\frac{P \times (1-P)}{N+1}} \; given \; P = \frac{|A > B|}{N+1} \quad N = |A > B| + |B > A| \tag{4.3}$$

In general, it can be said that, LM can have a value (i) close to 1.0 when there is a strong succession relation between A and B, (ii) in the neighborhood of 0.5 when there exists an equal probability for both a succession between A and B and between B and A and (iii) zero when there exists no succession relation between A and B as stated in [3].

***Definition* (support for from-to chart, *suppFTC*).** Support for from-to chart is the ratio of transitions that are from predecessor *A* to successor *B*, to the total number of process instances at the training dataset (i.e. #*L*), as given in Equation 4.4.

$$suppFTC(A,B) = \frac{|A > B|}{\#L} \tag{4.4}$$

Due to the effect of #L parameter, suppFTC metric evaluates the overall frequency of activities A and B. Similarly, the *global metric* (GM) given in [3] aims to measure similar global effect through Equation 4.5:

$$GM = \left(|A > B| - |B > A|\right) \times \frac{\#L}{\#A \times \#B} \tag{4.5}$$

***Definition* (modified lift, ML).** Thresholds defined over support and confidence metrics are to be parameterized by process engineers. Thus relatively lower confFTC and suppFTC threshold values may result in overfitting. To tackle this problem, a correlation measure called modified lift, which is calculated as given in Equation 4.6, can be used to augment the support-confidence framework.

$$ML(A,B) = \frac{|A > B| \times \#G}{|A > *| \times |* > B|} \tag{4.6}$$

In this formula, |A > B| is the total number of transitions from activity *A* to activity *B*, |A >*| is the total number of transitions initiated by activity *A* (i.e. row total of activity *A* at from-to chart), |* >B| is total

number of transitions attained to activity *B* (i.e. column total of activity *B* at from-to chart), and #*G* is grand total of scores at the from-to chart. According to modified lift value, scores in from-to chart are interpreted and processed as follows:

i. If the modified lift value is *greater than 1*, this implies that activities *A* and *B* are *positively correlated*, meaning that the occurrence of activity A potentially triggers the occurrence of activity B. Thus the score at element (A,B) in from-to chart does not change.
ii. If the modified lift value is *equal to 1*, this implies that activities *A* and *B* are *independent* and there is no correlation between these two activities. Thus the score at element (A,B) in from-to chart is reset to zero.
iii. If the modified lift value is *less than 1*, this implies that activities *A* and *B* are *negatively correlated*, meaning that the occurrence of activity *A* discourages the occurrence of activity *B*. Thus the score at element (A,B) in from-to chart is multiplied by -1. This negative factor is defined as Big M method in linear programming [75].

4.2. Genetic Algorithms Adaptation at Dominant Behavior Extraction

Genetic Algorithms (GA), which is a type of evolutionary algorithms, became popular by the research of John Holland in the early 1970s. It is a search algorithm that aims to find the best or approximate solutions for optimization or search problems. In this algorithm, a solution set is called a *chromosome*, and a value in a solution set is called a *gene*. In order to use genetic algorithm over a problem, the problem has to satisfy the following characteristics [76]:

- The final solution obtained using GA should not be expected to be the global optimal.
- The solution (i.e. chromosome) should consist of a series of values (gene), and every solution should be of the same length.
- The intermediate solutions should easily be evaluated according to the problem.
- The set of all possible solutions must be known clearly by the system, and a subset of solutions should easily be generated even if they are far from being the best solutions to the problem.

The power of GA depends on the origins of evolution theory [77]. By simulating the genetic process in real life, GA are able to evolve the solutions to dominant behavior extraction by selecting the strongest individuals and mating them, if they are appropriately encoded. Indeed, a problem might have several peak points (local optimal) in search space. Unlike to other *myopic local search algorithms* (e.g. hill-climbing search), GA can abandon inefficient local optimal by the undirected jumps triggered by crossover and mutation stages [77].

In this study, GA engine component that is adapted to dominant behavior extraction phase aims to find the dominant behavior within the process with the minimum total moment value (Z objective function given in Equation 4.1) in from-to chart. Unlike the prior permutative (brute-force) approach introduced in [42], which attempts to traverse all search space and is burdened with *quadratic assignment problem* (QAP), GA based dominant behavior extraction iteratively searches the global or sub-global optimum without exhausting the solution space in a parallel process starting from a set of feasible solutions (population) and it generates the candidate solutions in random fashion [78]. Although permutative approach is highly-dependent to the process complexity (i.e. the number of activities), GA relaxes this dependency, lowers the computational complexity and diminishes the total processing time intervals to practical and feasible levels. The basic GA stages are as follows:

i. *Initialization*. In this stage, an initial population is generated. This generation can be done in two different ways: random selection of the initial population and selection of potential individuals that satisfy the *schema*. According to Holland's schemata theorem [79], it is a pattern of gene values that may be represented by a substring of characteristics. It is assumed that an individual's high fitness (or high probability for mating) is due to the fact that it inherits good schemata.
ii. *Fitness Score Calculation.* Fitness score, i.e. *f(z)*, returns a single numerical fitness or figure of merit, which is in proportion to the utility or ability of the underlying chromosome to solve the problem [79].
iii. *Selection*. Selection stage is where the evolutionary theory steps in. In this stage, the successive population is generated by selecting individuals from the current population using a philosophy that is based on including the better individuals (*survival of the fittest*).
iv. *Crossover*. The major point in GA design is the balance between two opposite forces: while selection aims to shrink the diversity of population by unifying the content of the corresponding population, crossover and mutation attempt to increase the diversity of population by indirect jumps at the search space [78]. Additionally, the initial population is quite diverse early in the process, so crossover frequently takes larger steps in exploring the search space early in the search process and smaller steps later on when most individuals are quite similar [77].
v. *Mutation*. Mutation randomly alters each gene value at the offspring chromosome with relative small probability (typically with P(M)=0.02). In higher order domain alphabets, e.g. facility layout problem and 8-queen problem [77], in which binary coding is not appropriate, mutation takes the form of altering the current gene value with a random value that is chosen from the gene range with the mutation probability [80].

CHAPTER 5

PROPOSED APPROACH

Within the scope of this study, we aim to present a cross-organizational process mining framework that can identify commonalities among the organizations performing essentially the same process and to highlight the evolution of corresponding process alternatives (variants) according to duplication or divergence from these common regions. Underlying framework consists of three phases: *dominant behavior extraction*, *sequence alignment* and *process configuration*. Dominant behavior extraction phase initially derives the representative exemplary sequence that reflects the dominant behavior of the process, i.e. a typical or common intended behavior that can constitute the backbone of the process.

As the second phase, alignments among discovered dominant behaviors are performed by two different techniques: *Multi-Sequence Alignment* (MSA) and *Pairwise Alignment* (PA) both of which are Needleman-Wunsch algorithm adaptations with dynamic cost functioning. Dynamic cost derivation for edit operations within alignment is based on *confidence values*, which is the frequency of consecutive pair of activities sharing an incorporated business context in the event logs. While in Multi-Sequence Alignment technique, the intermediate pairwise alignments are combined together following a dendrogram-like structure namely *process family tree*, various clustering algorithms (e.g. K-Means, expectation maximization (EM) and agglomerative hierarchical clustering (AHC)) are applied at Pairwise Alignment technique to determine the underlying process families, which are the clusters of process alternatives sharing similar functionality and business context.

As the final phase in process configuration, multi-sequence or pairwise alignments among process variants are visualized at the alignment matrices. The functional inheritance among the process variants is interpreted by the feature sets namely *identical* and *maximal identical pairs* (IP and maxIP). The overview of proposed approach is given in Figure 5.1.

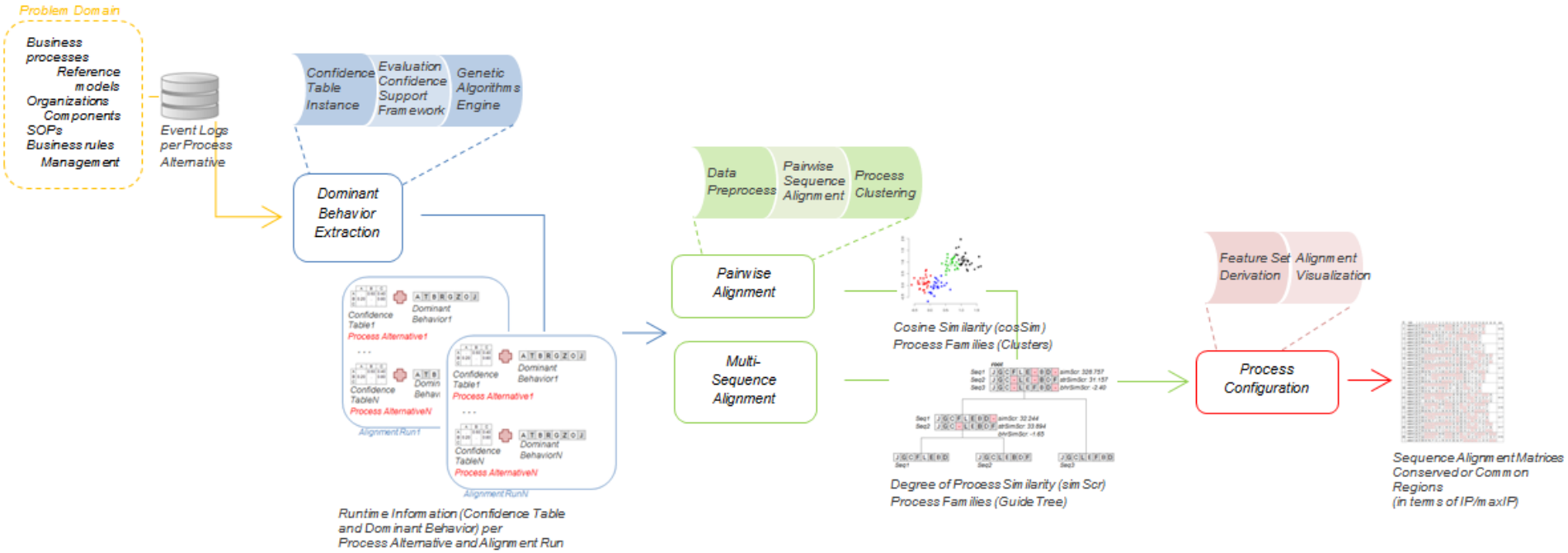


Figure 5.1. The Overview for Proposed Three-Phased Cross-Organizational Process Mining Framework. This framework consists of three phases: *dominant behavior extraction*, *sequence alignment* and *process configuration*.

5.1. Dominant Behavior Extraction

This phase aims to perform transformation from the event logs to the dominant behavior. The event logs are often referred to as history or audit trail, which typically contain the behavioral information about events assigned to an activity and process instance [10]. Dominant behavior is the most common and typical behavior that is embedded in multi-set of event logs. Data transformation in dominant behavior extraction phase is depicted in Figure 5.2.

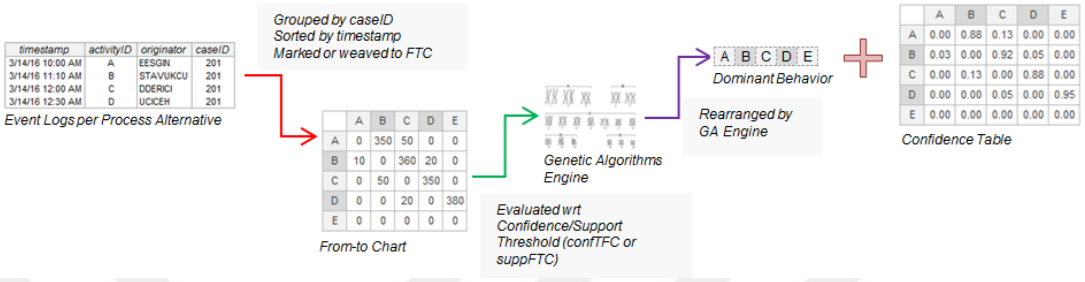


Figure 5.2. Data Transformation from Event Logs to Dominant Behavior. The multi-set of event logs is converted into single line dominant behavior with respect to many-to-one (*N:1*) cardinality. The operational-level interactions among activities and activity enabling are summarized at *from-to chart* and these values are converted to *confidence tables*.

5.1.1. The Concept of Dominant Behavior

When process observers attempt to compare business processes, they mostly take the graphical structures of the corresponding processes into account. They check whether the tasks occur in both process models and have similar successive connections or not [17, 20]. While informal process models, which are not designed in term of Business Process Modeling Notations (BPMN), are too imprecise and unclear, formal modeling languages are hard to understand by the process observers [50]. Additionally, process mining algorithms tend to discover complex spaghetti-like process models that are hard to comprehend while handling unstructured environments with concept drift [21].

Since there happens significant discrepancies between reference process structures and process behavioral patterns, it sounds sensible to measure the process equivalence according to actual process executions, which are defined by *firing sequences* (activity enabling sequences) emphasized at event logs [17, 20, 50]. The classical approach is to derive a state space or to enumerate all possible process streams and then compare the candidate models based on these abstractions. *Trace equivalence* and *bisimulation* are typical equivalence notions used for comparing formal models on such basis [20, 31]. Unfortunately, these techniques are only valid for the process models with formal description within an appropriate semantic and finite behavior which can be solidified by the number of traces or states [20, 31]. Moreover such notions provide an atomic true-false answer. In reality there will seldom be a perfect fit. Partial similarities in full firing sequences (i.e. partly-fitting sequences) should also be scored [81]. Indeed, it is focused on *the degree of similarity*, i.e. a continuous value between 0 (quite distinct) and 1 (totally similar). Respectively, prior measurement metrics tend to interpret all sub-processes and events as if they are equally important and probable. But there should be a balance between rarely active and significant fragments of the process model [82], likewise the *process vein* and *process arteries* analogy in [20]. Equivalence notions stated above are summarized in Table 5.1.

Table 5.1. Advantages and Disadvantages of Model-Model and Log-Model Equivalence Notions [20, 31].

Equivalence Notion	Advantages	Disadvantages
Equivalence of processes based on model-model similarity	*- Simple. It is checked majorly whether the same activities appear in both models and do they have similar connections.*	*- The graphical structure may be misleading.*
Equivalence of processes based on log-model similarity	*- It is focused on quantifying the equivalence of processes based on their process behaviors.*	*- The set of full firing sequences needs to be finite.* *- It does not take into account differences in importance such that a single deviation in a firing sequence invalidates the entire sequence.* *- The moment of choice is partially taken into account.*

Instead of prior process model-based approaches in measuring similarities between business processes, in this study we propose a quantitative approach that is based on common subsequence of behaviors captured in the event logs. Such repetitive behaviors are expected to occur within a process instance or across process instances and reflects some domain significance in terms of major use cases given in Section 4.1.1. Hence business observers can learn whether there are interesting execution patterns in the event logs. This new perspective to process mining uncovers the deviations and similarities by analyzing these common behaviors, thereby avoiding the requirement for well-structured process models. This most common (likely) and typical behavior obtained on the basis of event logs is figured out by a representative sequence called *dominant behavior*.

Definition **(dominant behavior)**. Dominant behavior is the sequence of activities with the length of *m*, $<a_1, a_2 \ldots a_m>$, that satisfies the objective function given in Equation 4.1. There exists an indexing function *ind:* $a_i \rightarrow$ {A, B…N} that maps i^{th} activity at dominant behavior to a unique activity label in activity type vocabulary (domain) with the size of *n*.

$$Min \sum_{i=1}^{m} \sum_{j=1}^{m} f_{ai,aj} \times |j-i| \times p$$

$$s.t.$$

$$p = \begin{cases} 1 & if\ i \leq j \\ > 1 & o/w \end{cases}$$

$$f_{ai,aj} \geq 0$$

$$ind : a_i \rightarrow \{A, B \ldots N\}\ where\ 1 \leq i \leq n$$

While the underlying linear program (LP) tends to assign the activity couples with relatively strong interactions, i.e. $f_{ai,aj}$ total number of transitions from predecessor a_i to successor a_j, to neighboring positions in the dominant behavior. Potential backtrackings such that $j>i$, are given a high penalty point *p*, which is parameterized at dominant behavior extraction. As a result, the *compactness* (i.e. the tendency towards more straight-line or direct type use cases emphasized in Section 4.1.1) of the underlying activity sequence is improved. Indeed, the dominant behavior handles the enabled transitions emphasized within event logs and considers these process behaviors, i.e. it is not just validated whether a task in a process trace is probable, but its successive relations within the corresponding process trace are also taken into account.

Hence the following enhancements are addressed by the concept of dominant behavior:

- Prior equivalence notions are only applicable for the process models that are designed by formal modeling notations and represent finite behavior [30]. Moreover, an atomic response rather than a degree of similar is required [31]. It is aimed to measure the similarity on a continuous basis.

- Partial fits is applicable and valuable. i.e. small local deviations between process models do not imply a complete *missfit*.
- The moment of choice in terms of succession firing at branching of process models is considered, since the goal is the task enabling shown by $f_{ai,aj}$ at dominant behavior formulation.

Dominant behavior extraction phase consists of two steps: (i) Data transformation and filtering, (ii) Genetic Algorithms based dominant behavior extraction.

5.1.2. Data Transformation and Filtering

The starting point of dominant behavior extraction phase is the instantiation of the from-to chart by retrieving all enabled activity labels from the event logs. Basically, event log dataset consists of four major attributes: *activityID*, *caseID*, *originator* and *timestamp*. Figure 5.3 exemplifies a sample set of event logs involving 9 events.

Timestamp		Originator	Activity	CaseID
9/27/2015	14:25:10	PatrickDelfmann	A	172
9/27/2015	14:25:13	RenLu	C	172
9/27/2015	14:25:16	AdrianBuijs	D	172
9/27/2015	14:25:19	ThomasBauer	G	172
9/27/2015	14:25:22	SmithBetz	F	172
9/27/2015	14:25:25	AdrianBuijs	B	172
9/27/2015	14:25:28	RenLu	I	172
9/27/2015	14:25:31	SmithBetz	H	172
9/27/2015	14:25:10	SteveRinderle	A	173
9/27/2015	14:25:13	PatrickDelfmann	C	173
9/27/2015	14:25:16	AlenaHallerbach	D	173

Figure 5.3. Event Logs in the form of *<timestamp, originator, activityID, caseID>* for Travel Management Process with *caseID=172* and *173*.

For populating the from-to chart table, event logs are grouped by process instances and then ordered by timestamp in ascending order. Hence transaction streams that comply with the original execution are constructed. Then, predecessor (*P*) and successor (*S*) tasks are parsed for each transition in transaction streams and the current score of $(P,S)^{th}$ element at the from-to chart is incremented iteratively [66]. As a result, all transitions among activities in process instances are recorded at from-to chart.

Then total scores at from-to chart are analyzed by the evaluation metrics given in Section 4.1.2: *confidence for from-to chart* (confFTC) and *support for from-to chart* (suppFTC). Such an evaluation step is required to prune weak scores prior to dominant behavior extraction and eliminate their effect on the fittest activity sequence as stated in [42]. Figure 5.4 summarizes the event log transformation and filtering steps.

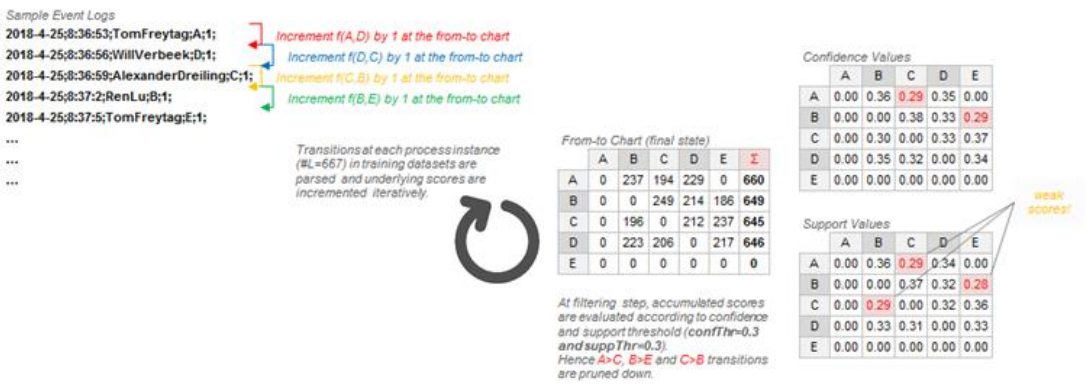


Figure 5.4. Event Logs Transformation and Filtering Steps.

5.1.3. Genetic Algorithms Based Dominant Behavior Extraction

This operation is the engine component of dominant behavior extraction phase, which aims to construct the consensus activity sequence with the minimum total moment value (*Z* in Equation 4.1) at the from-to chart. As stated in Section 4.2, this optimal (or sub-optimal) activity sequence conceptualizes typical intended behavior that is found to recur within process instances with some domain significance. The pseudo code for GA-based dominant behavior extraction is given in Algorithm 5.1:

Algorithm 5.1: GA-based Dominant Behavior Extraction(*from-to chart*)

```
1:  REFRESH schema, initPopulation, currPopulation, newPopulation, domBeh
2:  if schemaApplied IS TRUE then
3:    CONSTRUCT schema
4:    GENERATE initPopulation WITH schema
5:  else
6:    RANDOM GENERATE initPopulation
7:  endif
8:  COMPUTE FITNESS for initPopulation
9:  COPY initPopulation TO currPopulation
10: INIT I
11: while i ≤ maxIteration do
12:   while size of newPopulation  < maxSize do
13:     SELECT parent1 AND parent2 FROM currPopulation
14:     [child1, child2] ← CROSSOVER parent1 AND parent2
15:     MUTATE [child1, child2]
16:     ADD [child1, child2] TO newPopulation
17:   endwhile
18:   COMPUTE FITNESS for newPopulation
19:   COPY newPopulation TO currPopulation
20:   CLEAR newPopulation
21:   if currPopulation IS CONVERGED then
22:     TERMINATE
23:   endif
24:   INCREMENT i by 1
25: endwhile
26: dominant behavior ← SEARCH currPopulation WITH MAX(fitness)
```

The mapping of basic GA notations into the business process modeling domain is as follows; a candidate solution set (i.e. *chromosome* or *genotype* encoding the dominant behavior) possessed by an individual is represented as an *activity sequence* and each value in this solution representation (i.e. *gene*) corresponds to a unique *activity label*. Finally, the genetic information encapsulated by chromosome is converted to an organism or *phenotype* (e.g. business process model). Figure 5.5 depicts the basic GA notations adapted to business process modeling domain.

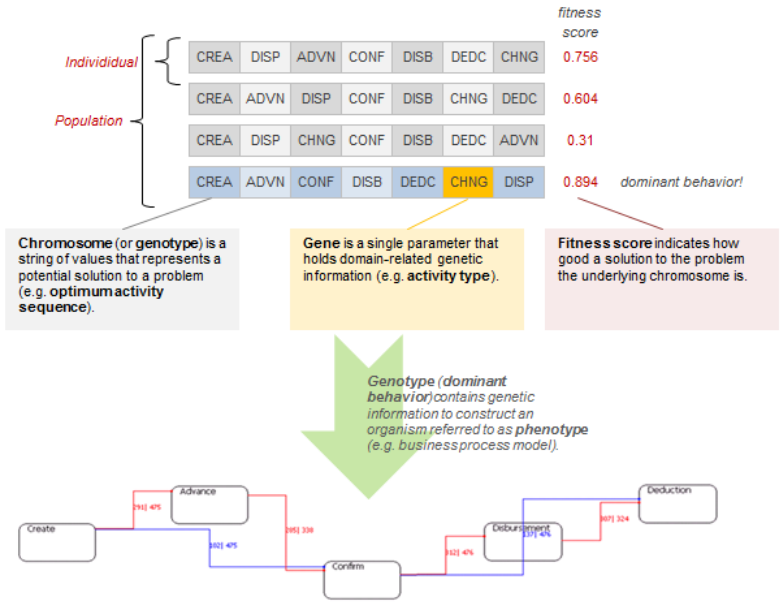


Figure 5.5. Mapping of basic GA Notations into Business Process Modeling Domain.

The GA stages stated in Section 4.2 are adapted as follows:

Initialization. In the case of schema application, scores that are recorded at from-to chart are retrieved and sorted in descending order. Then a top-down search is performed to create a non-intermittent schema with a predefined maximum length[2]. Such maximal limit for the length of schema is important, since relatively longer schema may restrict the distribution of initial population at a certain portion of the search space and it would be less probable for indirect jumps to find out alternative solution paths. A sample initialization is shown in Figure 5.6.

Fitness Score Calculation. As far as GA are concerned, it is preferred to maximize a given better fitness score in order to provide more opportunities especially in selection stage. Therefore *the inverse of the objective function* given in Equation 4.1 is used as the denominator of the fitness function to search for the solution with the minimum value. The numerator of the fitness function is set to the total scores in the from-to chart.

$$f(z) = \frac{\sum_{i=1}^{|chrZ|}\sum_{j=1}^{|chrZ|} score_{ij}}{\sum_{i \in chr\ Z}\sum_{j \in chr\ Z} score_{ij} \times |j-i| \times p} \qquad (5.1)$$

According to Equation 5.1, the best fitting individuals are selected according to the fitness function *f(z)* that aims to maximize the compactness of the dominant behavior by favoring the activities with higher scores to adjacent (neighboring) positions. Due to the moment notation, the maximum value for fitness function is theoretically 1.0, i.e. all non-zero scores at from-to chart are aligned just above the diagonal.

Selection. As the selection method, *roulette wheel selection* is applied. Roulette wheel selection is a kind of random selection type where individual *i* has a probability of $f_i(z)/\sum f(z)$ to be selected as a parent to mate. Since higher fitness score means higher chance to mate, the random choice is biased towards the fitness score.

[2] *Non-intermittent schema* implies that the underlying schema does not include undefined (#) gene value.

Crossover. In our solution, crossover is not applied to all pairs of parents selected for mating. The default likelihood of crossover is set as P(C)=0.8. If the crossover is bypassed, the offspring are produced by simply duplicating the parents. Otherwise, a random crossover gene position is selected and chromosome subsets are exchanged according to this position.

Mutation. Conventional mutation and crossover framework may cause problems with *chromosomes legality*, e.g. multiple copies of a given activity type may occur at the offspring. Therefore we propose an alternative mutation scheme that automatically swaps the duplicate gene value with a randomly selected unobserved one. Hence a uniform chromosome that satisfies the chromosome legality is reproduced.

Population Convergence. As a termination condition, if at least 95% of the individuals at the current population are in the *convergence band*[3], no more new population should be generated. In order to promote the *premature convergence*, convergence ratio parameter has to be determined appropriately. Finally, gene sequence (genotype) of the individual with the maximum fitness score at the last population decodes the dominant behavior. Figure 5.6 gives a sample run for GA-based dominant behavior extraction phase.

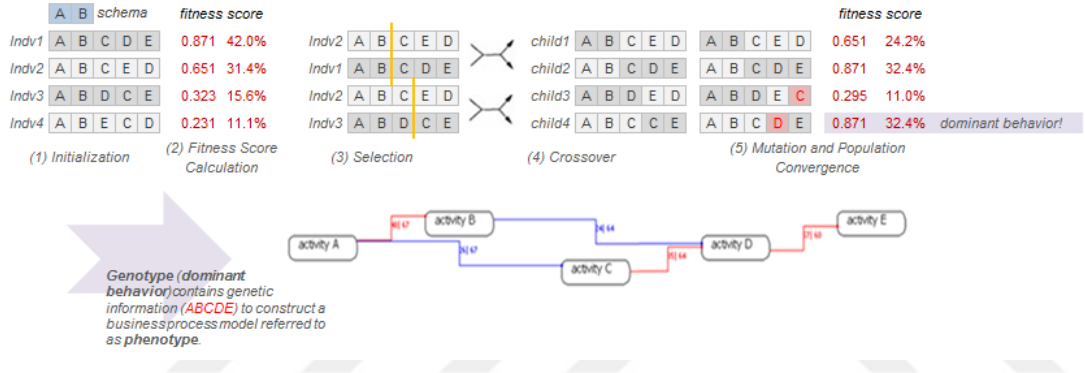


Figure 5.6. A Sample Run for GA-based Dominant Behavior Extraction. The *genotype* (ABCDE), i.e. dominant behavior, constitutes the backbone for the *phenotype*, i.e. business process model.

[3] The interval of *convergence band* is delimited as [(1-*convergence ratio*), 1].

5.1.4. Design of Experimental Runs for Dominant Behavior Extraction

Dominant behavior extraction phase is configured according to two sets of parameters: *process discovery parameters* and *Genetic Algorithms parameters*. Table 5.2 summarizes the underlying runtime parameters and their features:

Table 5.2. Dominant Behavior Extraction Runtime Parameters.

Group	Parameter	Description
Process Discovery Parameters	*Confidence Threshold*	*Threshold value for confidence metric (confFTC) introduced in section 4.1.2.*
	Support Threshold	*Threshold value for support metric (suppFTC) introduced in section 4.1.2.*
	Eliminated Closed-Loops	*A boolean parameter (yes/no) to determine whether one-step closed loops at process discevory will be eliminated or not.*
	Backtracking Penalty Point	*Backtracking penalty point (p) given at Equation 4.1. It is used to enforce dominant behavior extraction phase to minimize the loop-backs at process discovery.*
	Verification Method	*The verification method (e.g. hold-out or N-fold cross validation) applied to measure completeness and soundness.*
Genetic Algorithms Parameters	*GA Applied*	*A boolean parameter (yes/no) to determine whether GA or random permutation is applied.*
	Schema Application	*A boolean parameter (yes/no) to determine whether schema is applied at initial population generation or not.*
	Population Size	*Population size for GA implementation at dominant behavior extraction.*
	Number of Elite Individuals	*Number of elite individuals (i.e. individuals with relatively higher fitness score at the current population) that are directly passed to next population (or iteration).*
	Convergence Ratio	*Convergence ratio used to design the size of convergence band introduced in section 5.1.3. If at least %95 of the individuals reside in the convergence band, GA iterations are terminated.*
	Maximum Iteration	*Maximum iteration number for GA implementation at dominant behavior extraction.*
	P(crossover)	*Likelihood of crossover operation in [0,1] interval.*
	P(mutation)	*Likelihood of mutation operation in [0,1] interval.*

According to *tacit process variant assumption*, which states the fact that there is no available knowledge on how to partition the set of cases [48], there may arise an inductive biasness at dominant behavior extraction. To minimize this occurrence, *N* consecutive runs with varying process discovery and Genetic Algorithms parameter settings are performed and *N* distinct or quite similar versions of dominant behaviors per process alternative are extracted. In the following section, an illustrative example for dominant behavior extraction phase is given.

An Illustrative Example for Dominant Behavior Extraction. As the starting point, 3 synthetic process variants are considered. 1000 process instances per process variant are synthetically generated according to the reference process models given in Figure 5.7.

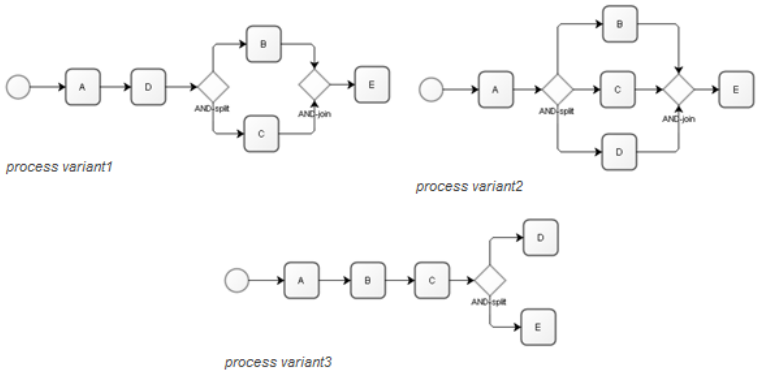


Figure 5.7. Process Models per Synthetic Process Variant (*process variant1–3*).

Then 5 sample runs are configured and performed according to the process discovery and Genetic Algorithms parameters given in Table 5.3. At the corresponding uses cases handled in Experimental Analysis Chapter, the number of sample runs is determined according to the cartesian multiplication of the runtime parameters given in Table 5.2. Figure 5.8 shows the user interface of ProMiner software, which is used to discover process patterns from event logs in the form of dependency/frequency and control flow graphs. This program was developed in the scope of the author's Master of Science dissertation [83].

Table 5.3. Process Discovery and Genetic Algorithms Parameters Configuration per Process Discovery Run. In order to eliminate potential inductive biasness, underlying runtime parameters are configured according to process engineers and domain expert's feedbacks.

runID	1	2	3	4	5
event log size	1000	1000	1000	1000	1000
confidenceFTC	**0.3**	**0.15**	**0.3**	**0.5**	**0.3**
supportFTC	**0.3**	**0.15**	**0.3**	**0.5**	**0.3**
backtrack. penalty point	2	2	2	3	2
eliminate closed loops	*yes*	*yes*	*yes*	*no*	*yes*
verification method	*holdout*	*holdout*	*holdout*	*holdout*	*holdout*
GA applied	*yes*	*yes*	*yes*	*yes*	*yes*
population size	80	80	100	80	50
number of elite indv.	8	8	10	8	5
convergence ratio	0.15	0.15	0.15	0.15	0.15
max. iteration	200	200	200	200	300
P(crossover)	0.8	0.8	0.5	0.8	0.8
P(mutation)	0.02	0.02	0.1	0.02	0.02
apply schema	*yes*	*yes*	*yes*	*yes*	*yes*

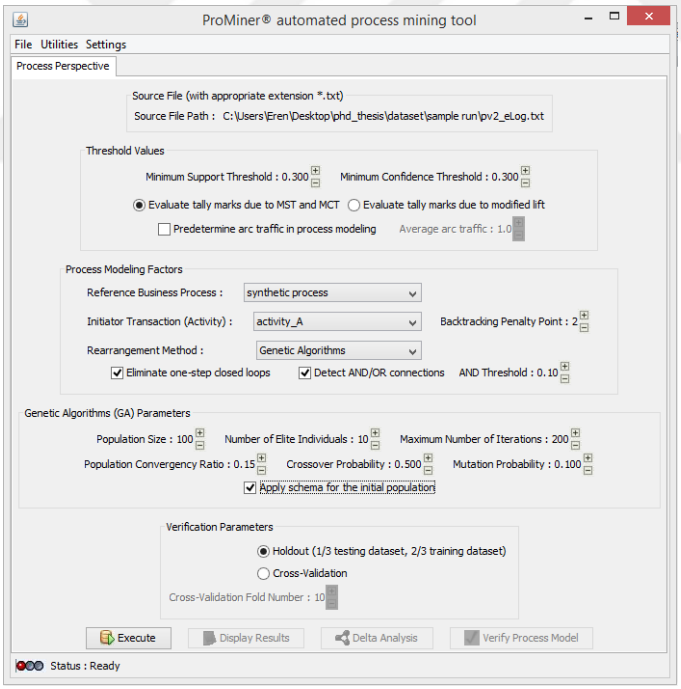


Figure 5.8. User Interface of ProMiner Software. As an example, process discovery and Genetic Algorithms parameters are configured for *variant2* and *process discovery run=2*.

As the output of this phase, dominant behavior in the form of consensus activity sequences and final states of from-to chart that summarizes the interactions among activities are generated as shown in Table 5.4 and 5.5 respectively. Additionally, each process discovery run is evaluated according to the following attributes:

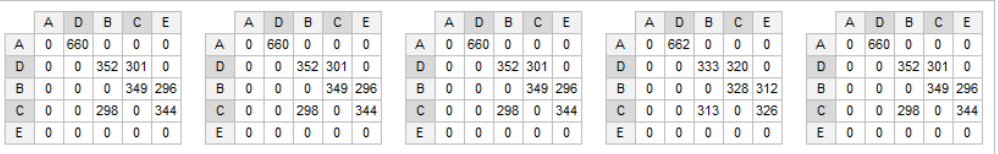
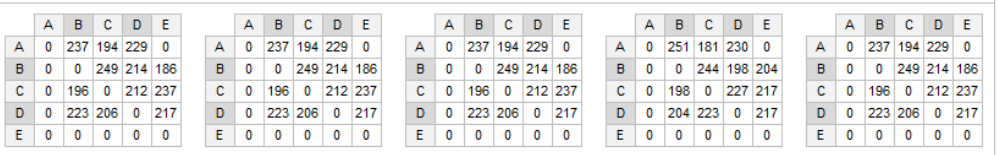
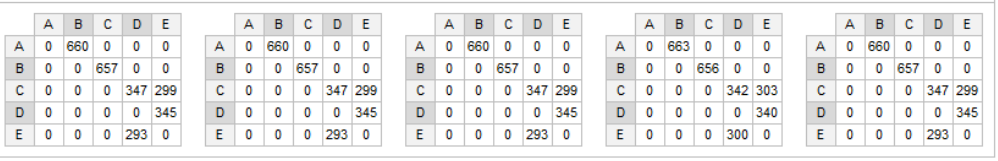
- *Completeness*. This is the fraction of the traces in the event log that may be the result of some enactment at the corresponding process model. It is relevant with accuracy aspect of process discovery.

- *Soundness*. Soundness measures the fraction of the enactments at the corresponding process model *PM* that find some correspondence in the event logs. It is also similar to minimality, behavioral appropriateness or precision. Details about completeness and soundness metrics are given in Section 6.3.1.
- *Average transition length*. Average transition length measures the compactness of the process discovery in terms of transition length. According to the fitness function given in Equation 5.1, dominant behavior extraction enforces GA engine component to assign the activities with relatively strong interdependency to adjacent (neighboring) positions at dominant behavior.
- *Average transition number per activity*. This measure is related to the effectiveness of confidence and support threshold framework in filtering relatively weak interactions. Process discovery with higher average transition number may tend to be spaghetti-like process model.
- *Total processing time*. Total processing time is the total cycle time of a single process discovery run. This cycle time is directly proportional to the size of training dataset, population size, maximum iteration number and verification method selection (e.g. N-fold cross validation with extremely high fold number N).

Table 5.4. Runtime Information per Process Discovery Run. Each process discovery run is evaluated according to completeness, soundness, average transition length, average transition number per activity and total processing time attributes.

	runID	1	2	3	4	5
dominant behavior	*variant1*	A D B C E	A D B C E	A D B C E	A D	A D B C E
	variant2	A B D C E	A B D C E	A B D C E	A	A B D C E
	variant3	A B C D E	A B C D E	A B C D E	A B C D E	A B C D E
completeness	*variant1*	87.841	87.481	87.481	25.229	87.481
	variant2	59.754	75.499	59.754	0.00	59.754
	variant3	90.382	90.382	90.382	72.26	90.382
soundness	*variant1*	56.836	56.836	56.836	99.099	56.836
	variant2	33.276	32.701	33.276	NA	33.276
	variant3	62.891	62.891	62.891	70.090	62.891
avg transition length	*variant1*	1.33	1.33	1.33	1.00	1.33
	variant2	1.43	1.78	1.43	0.00	1.43
	variant3	1.29	1.29	1.29	1.00	1.29
avg transition number per activity	*variant1*	1.20	1.20	1.20	0.83	1.20
	variant2	1.40	1.80	1.40	0.00	1.40
	variant3	1.17	1.17	1.17	0.83	1.17
total processing time (sec)	*variant1*	20	19	20	15	20
	variant2	19	19	20	15	23
	variant3	22	22	23	15	22

Table 5.5. Final State of From-to Chart per Process Discovery Run. These from-to chart instances are crucial for calculating confidence values (*confFTC*) at sequence alignment phase.

runID	1	2	3	4	5
variant1					
variant2					
variant3					

In addition to this runtime information, ProMiner software converts the discovered process knowledge in two distinct forms: (i) *dependency/frequency graph* and (ii) *control flow graph*.

5.2. Sequence Alignment

There is a well-known metaphor in biology; if any two protein sequences (e.g. DNA, RNA) are similar, they have also similar functions or 3D structures [22]. *Sequence analysis* in bioinformatics domain often compares this similarity between two biological sequences to understand their structures or functionalities. Some sample applications of sequence analysis are predicting the biological function of a gene, finding the evolution distance, a common region in two genomes or repeats within a genome. For example, *tandem repeats* are related to various mechanisms such as protein binding [23].

Similarly, in process diagnostics, common subsequences of activities in event logs that are found to repeat within a process instance or across process instances highlight some domain significance such as direct succession between the activities with relatively high inter-dependency. Additionally, the discrepancies, exceptional behavior and niche-type events are handled according to the reference processes and we employ a sequence alignment mechanism in order to compare the process variants for spotting the similarities by analyzing the dominant behavior.

5.2.1. Preliminaries on Sequence Alignment

Needleman-Wunsch (*NW*) *algorithm*, using the dynamic programming approach, aims to find the global optimal alignment between two amino acid sequences [22]. The basic motivation of NW algorithm is to generate a global optimal alignment by progressively calling the previous solutions that optimize the alignment of smaller subsequences [23]. The challenge is to find an alignment that is as simple and informative as possible. Rather than local optimal alignment algorithms, e.g. Smith-Waterman, we focus on the global optimal alignments because of the following reasons:

- The requirement for handling of the whole process execution. Since local optimal alignments only handle a fragment of the dominant behavior, it is not suitable for finding common patterns that can converge the entire trace of the underlying process execution.
- Alignment shrinkage due to the noisy event logs. The common fragments tend to be short due to the noise at the event logs. This rationale may propagate into the alignment shrinkage for the local optimal alignment algorithms with non-informative shorter commonalities detected among the process alternatives.

Let T_1 and T_2 be two sequences, namely *source* and *target* sequences. *Needleman-Wunsch matrix* (*F*) indexed by *i* and *j*, is constructed where the value $F(i,j)$ is the score of the best alignment between the prefix T^i_1 of T_1 and the prefix T^j_2 of T_2. $F(i,j)$ is initialized by $F(0,0)=0$ and then proceeds to fill the matrix from top left to bottom right. It is possible to calculate $F(i,j)$ according to neighboring values, $F(i-1,j)$, $F(i-1,j-1)$ and $F(i,j-1)$. There are three possible ways that the best score $F(i,j)$ of an alignment up to subsequences T^i_1 and T^j_2 can be obtained as given in Equation 5.2.

$$F(i,j) = max\begin{cases} F(i-1,j-1) + M\left(T_1^i, T_2^j\right) \\ F(i-1,j) + I_2\left(T_2^j, T_1^i, T_2^{j+1}\right) \\ F(i,j-1) + I_1\left(T_1^i, T_2^j, T_1^{i+1}\right) \end{cases} \tag{5.2}$$

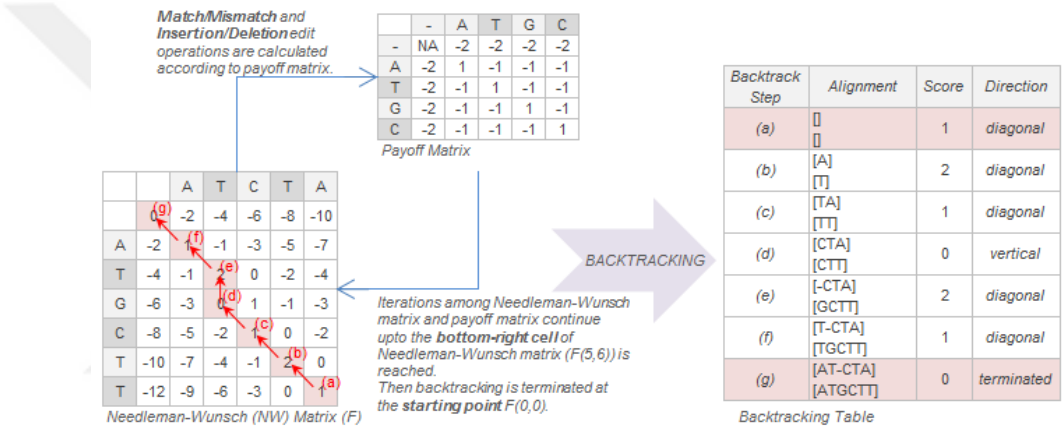
In Equation 5.2, $M(T^i_1, T^j_2)$, $I_1(T^i_1, T^j_2, T^{i+1}_1)$ and $I_2(T^j_2, T^i_1, T^{j+1}_2)$ refer to *Match/Mismatch* and *Insertion/Deletion* (i.e. inDel) edit operations, respectively. While these edit operations are valuated according to payoff matrices PAM or BLOSUM in bioinformatics literature [22], we use dynamic cost functioning which is based on confidence values extracted from confidence table generated at dominant behavior extraction phase. The basic idea of this *confidence enhanced cost functioning* is to associate the actual frequency of activity combinations that have common and specific business context according to the confidence values with the expected frequencies and then interpret whether they occur in a correlated and dependent fashion or not.

In order to consider the best alignment, we need a merit for associating similarity degree with the alignment. The value at the bottom right cell of the NW matrix, $F(|T_1|,|T_2|)$, is the similarity score, i.e.

simScr(T_1,T_2), for the alignment of sequences T_1 and T_2. In order to find out the optimal alignment, we must *backtrack* the path of choices by Equation 5.2 that led to this best score, i.e., we move from the current cell (*i,j*) to one of the neighboring cells from which the value *F*(*i,j*) is calculated. At backtracking step, pair of symbols is added to alignment as follows:

- T^i_1 and T^j_2 if the move is to (*i-1,j-1*). This denotes the substitution of activity labels among sequences T_1 and T_2.
- T^i_1 and the gap symbol – if the move is to (*i-1,j*). This denotes the deletion of activity T^i_1 at sequence T_1 and the insertion of activity T^i_1 to sequence T_2.
- The gap symbol – and T^j_2 if the move is to (*i,j-1*). This denotes the insertion of activity T^j_2 to sequence T_1 and the deletion of activity T^j_2 at sequence T_2.

Backtracking operation is terminated at the starting point (0,0). Figure 5.9 shows a sample alignment and backtracking procedure between two sequences, T_1 (`ATCTA`) and T_2 (`ATGCTT`). While NW matrix (F) holds the scores per iteration, backtracking table shows the optimal alignment through diagonal, vertical or horizontal moves.



	-	A	T	G	C
-	NA	-2	-2	-2	-2
A	-2	1	-1	-1	-1
T	-2	-1	1	-1	-1
G	-2	-1	-1	1	-1
C	-2	-1	-1	-1	1

		A	T	C	T	A
	0	-2	-4	-6	-8	-10
A	-2	1	-1	-3	-5	-7
T	-4	-1	2	0	-2	-4
G	-6	-3	0	1	-1	-3
C	-8	-5	-2	1	0	-2
T	-10	-7	-4	-1	2	0
T	-12	-9	-6	-3	0	1

Backtrack Step	Alignment	Score	Direction
(a)	[] []	1	diagonal
(b)	[A] [T]	2	diagonal
(c)	[TA] [TT]	1	diagonal
(d)	[CTA] [CTT]	0	vertical
(e)	[-CTA] [GCTT]	2	diagonal
(f)	[T-CTA] [TGCTT]	1	diagonal
(g)	[AT-CTA] [ATGCTT]	0	terminated

Figure 5.9. A Sample Alignment and Backtracking Procedure between T_1 and T_2 with 1.0 similarity score. Iteration and trace-back once iteration has been completed (*NW matrix* (*F*) at left-hand side) backtracking begins (*backtracking table* at right-hand side). During backtracking, the pairwise alignment between the two input sequences is constructed.

5.2.2. Multi-Sequence Alignment

Multi-Sequence Alignment is the progressive alignment technique that utilizes the adapted Needleman-Wunsch algorithm iteratively to achieve the multiple alignment of a set of dominant behavior sequences belonging to distinct organizations. Then it constructs *process families* depicting the relative distance and similarities among organizations. This iterative application of NW algorithm constructs a tree structure that shows process families structured according to the commonalities and differences among process variants. The two important features of this process family tree are its *topology*, or *branching order*, and its *branch length*, which ought to be proportional to normalized similarity score.

According to the *conceptual perspective*, Multi-Sequence Alignment technique is composed of two major data structures: (i) individual set and (ii) alignment run set.

- Individual set is a bag of *individual lists*, where each individual list represents all active individuals per unique alignment run (denoted by *alignmentRunID*).
- Each *individual* in an individual list consists of one or more *sequences* (i.e. at base level (*level=1*) an individual is composed of the original dominant behavior sequence, then aligned and combined forms of the underlying sequences at following higher levels) and a *level* attribute. This level attribute shows the level of process family tree at which the underlying individual is valid.

- *Sequence* holds the *element list* and a *process alternative* (or process variant) attribute. The process alternative indicates the ancestor organization to which underlying dominant behavior sequence belongs to.
- *Element* is an atomic entity that is composed of *original character* (i.e. the character used in visualizing process family tree) and an *indel character array*. This indel (insertion/deletion) character array holds the bag of characters that are inserted at the previous levels according to “once-a gap always-a gap” policy stated in [84, 85]. By this array, current element is able to inherit and imitate the behaviors of these previously inserted characters.
- *Alignment run set* is a bag of *alignment run lists*, which holds all performed pairwise alignments per unique alignment run.
- *Alignment* summarizes the characteristics of relevant alignment operation, i.e. source and target individuals (therefore it checks the validity and existence of the individuals at the underlying level), level and the similarity scores (e.g. similarity/normalized similarity scores and structural/behavioral similarity scores). In the case of *optimality* of the current alignment, a combined individual, which is a compound of aligned forms of source and target individuals, is created and passed to the next level as a new individual.

Figure 5.10 depicts the above mentioned structures of Multi-Sequence Alignment technique.

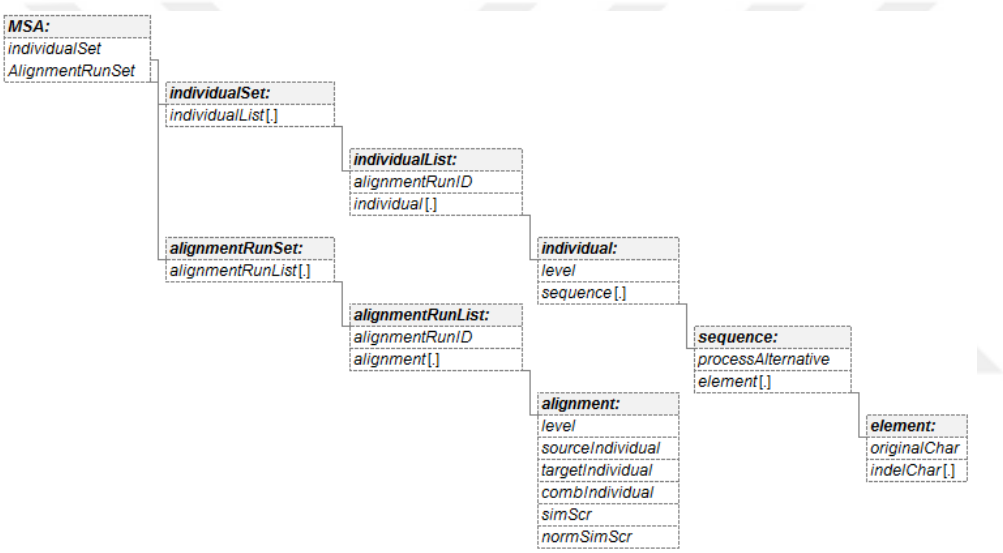


Figure 5.10. The Data Structure of Multi-Sequence Alignment (MSA) Technique. *MSA* object consists of two major data structures: *individual set* and *alignment run set*.

Similar to the “once-a gap, always-a gap” policy in CLUSTALW [84, 85], we consider the capability of multi-character inheritance within a gap symbol (–) due to the nature of progressive alignment. Figure 5.11 shows the details about the underlying behavior inheritance.

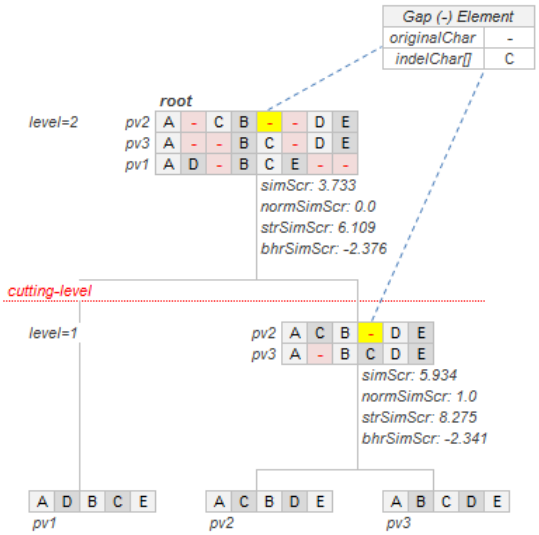


Figure 5.11. Gap Symbol Representation in Multi-Sequence Alignment. The gap symbol (–) is firstly inserted to *variant2* (*pv*$_2$) at *level=1* and this representation is retained at the following levels up to *root* node is left. *Cutting level* determines the set of clusters, namely *process families*. While *original character* attribute, which holds the character used in process family display, is set as gap (–), *indel character array* holds characters inserted at previous levels. Hence, gap symbol inherits and imitates the behavior of these previously inserted characters within inDel (insertion/deletion) edit operations.

Due to the use of multi-character inheritance within a gap symbol (–) in multi-sequence alignment, the *confidence metric* introduced in Equation 4.2 is revised as *average confidence* (*avgConfFTC*) denoted in Equation 5.3. In this equation, I^i_k stands for the element at the i^{th} position of sequence k that belongs to individual I, *ch* refers to the reference character or activity label. While *dir* is the direction of confidence computation, i.e. *f–forward*, *b–backward* and *i–insertion*, $|I^i_k|$ is the total length of indel character array of the underlying element.

$$avgconfFTC_{I_{individual\_index}}\left(I^{i\_element\ index}_{k\_sequence\ index}, ch, dir\right)=\begin{cases} dir: f \left(\sum_{x=1}^{\left|I_k^{i+1}\right|} confFTC\left(ch, I_k^{i+1,x}\right)\right) \Big/ \left|I_k^{i+1}\right| \\ dir: b \left(\sum_{x=1}^{\left|I_k^{i-1}\right|} confFTC\left(I_k^{i-1,x}, ch\right)\right) \Big/ \left|I_k^{i-1}\right| \\ dir: i \left(\sum_{x=1}^{\left|I_k^{i}\right|} confFTC\left(I_k^{i,x}, ch\right)\right) \Big/ \left|I_k^{i}\right| \end{cases} \quad (5.3)$$

The basic motivation of confidence-based cost functioning in Multi-Sequence Alignment is to interpret the actual frequency of activity pairs that have common business conditions and notated with the expected frequency of co-occurrence of activity *ch* if it occurs at the proposed direction, *dir*.

Calculating the Alignment Score for Combined Schema. Although one of the most popular scoring mechanisms for the multiple sequence alignment of genomic sequences is the *sum-of-pairs* (SP) [21, 62, 84], we prefer to generalize the dynamic programming paradigm of Pairwise Alignment approach to Multi-Sequence Alignment. Hence the generic objective function of Needleman-Wunsch algorithm given in Equation 5.2 is adjusted according to the conceptual perspective of Multi-Sequence Alignment technique as given in Equation 5.4:

$$F(i,j)=max\begin{cases} F(i-1,j-1)+\left(\sum_{k=1}^{|S|}\sum_{l=1}^{|T|} matchScr\left(S_k^i,T_l^j\right)\right)\Big/|S|\cdot|T| \\ F(i-1,j)+\left(\sum_{k=1}^{|S|}\sum_{l=1}^{|T|} indelScr_T\left(T_l^j,S_k^i\right)\right)\Big/|S|\cdot|T| \\ F(i,j-1)+\left(\sum_{k=1}^{|S|}\sum_{l=1}^{|T|} indelScr_S\left(S_k^i,T_l^j\right)\right)\Big/|S|\cdot|T| \end{cases} \quad (5.4)$$

Let *S* and *T* be two individuals or process variants (i.e. S stands for source and T for target individual) that compose one or more sequences at the current level (i.e. dominant behavior sequence at the base level (*level=1*), then aligned and combined forms of these sequences at following levels), $matchScr(S^i_k,T^j_l)$, $indelScr_T(T^j_l,S^i_k)$ and $indelScr_S(S^i_k,T^j_l)$ multipliers[4] in Equation 5.5 stand for *Match/Mismatch* and *Indel* (*insertion/deletion*) edit operators that are determined at *confidence enhanced dynamic cost function*. Dividing the total score by the Cartesian product of source and target individual's length (|*S*|.|*T*|) is similar to the scoring scheme in CLUSTALW [84].

In the case of matching ($S^{i,m}_k=T^{j,n}_l$, e.g. $S^{i,m}_k$ stands for m^{th} inDel character of element at the i^{th} position of sequence *k* that belongs to source (*S*) individual), a *normalized match score* is calculated via $normMatch(S^{i,m}_k,T^{j,n}_l)$ given in Equation 5.6. If the confidence values of current elements are highly significant (*avgConfFTC>>>confThr*) confidence-enhanced dynamic cost function tends to preserve this current pattern by assigning highly positive matching score. Dynamic cost function also assigns a default *confThr* value (confidence threshold) to the match case as shown in Equation 5.5.

On the other hand, if the current characters are different ($S^{i,m}_k\neq T^{j,n}_l$), an average *opportunity cost* ($oppCost_I(I^{i,m}_k,\bar{I}^{j,n}_l)$ given in Equation 5.7)[5] is calculated to measure the reactions of source and target individual to the replacement of current prefixes. Dynamic cost function initially assigns a default –*confThr* value to the mismatch case as shown in Equation 5.5. This value changes according to the outcome of the *opportunity cost* such that, the substitution of uncorrelated or contrasting elements is highly penalized by log_2 base, while substituting activities are encouraged to be replaced according to substantive business knowledge. Since the predecessor and successor of current prefixes are subjected at the opportunity cost, the idea behind this metric is similar to the flooding phenomenon of *3-gram distance* emphasized in [62].

$$matchScr\left(S_k^i,T_l^j\right)=\frac{\left(\sum_{m=1}^{|S_k^i|}\sum_{n=1}^{|T_l^j|}\begin{cases} normMatch\left(S_k^{i,m},T_l^{j,n}\right)+confThr & if\ S_k^{i,m}=T_l^{j,n} \\ avg\left(oppCost_S\left(S_k^{i,m},T_l^{j,n}\right),oppCost_T\left(T_l^{j,n},S_k^{i,m}\right)\right)-confThr & o/w \end{cases}\right)}{|S_k^i|\cdot|T_l^j|} \quad (5.5)$$

$$normMatch\left(S_k^{i,m},T_l^{j,n}\right)=-log_2\left(\frac{confThr^2}{\left(avgconfFTC_S\left(S_k^i,S_k^{i,m},f\right)\cdot avgconfFTC_T\left(T_l^j,T_l^{j,n},f\right)\right)}\right) \quad (5.6)$$

$$oppCost_I\left(I_k^{i,m},\bar{I}_l^{j,n}\right)=-log_2\left(\frac{\left(avgconfFTC_I\left(I_k^i,I_k^{i,m},b\right)\cdot avgconfFTC_I\left(I_k^i,I_k^{i,m},f\right)\right)}{\left(avgconfFTC_I\left(I_k^i,\bar{I}_l^{j,n},b\right)\cdot avgconfFTC_I\left(I_k^i,\bar{I}_l^{j,n},f\right)\right)}\right) \quad (5.7)$$

On the other hand, the *inDel* scoring is handled as the combination of two edit operations as denoted in Equation 5.8: *insertion* and *deletion*. As in the opportunity cost, confidence enhanced dynamic cost function assigns a default –*confThr* value to the inDel and this default value can be exaggerated by insertion and deletion of elements not conforming to business context. *Insertion cost* ($insCost_I(I^{i,m}_k,\bar{I}^{j,n}_l)$ given in Equation 5.9) compares the as-is situation with the relative cost of inserting the character from

[4] log_2 base at $matchScr(S^i_k,T^j_l)$, $indelScr_T(T^j_l,S^i_k)$ and $indelScr_S(S^i_k,T^j_l)$ parameters reflect exponentially decrease at cost functioning. This log-odds score effect is also emphasized in [21, 33, 34, 62].

[5] $\bar{I}$ stands for *non-I* individual, e.g. $\bar{S}$ implies *T*.

other individual ($\bar{I}^{j,n}_l$) between the current character ($I^{i,m}_k$) and its successor element (I^{i+1}_k). The potential cost of deletion character $I^{j,n}_l$ from individual I is handled by *deletion cost* ($delCost_I(I^{j,n}_k)$ given in Equation 5.10). Hence *business context driven inDel* scores are generated for each element at Needleman-Wunsch matrix.

$$indelScr_I\left(I_k^i,\bar{I}_l^j\right)=\frac{\left(\sum_{m=1}^{|I_k^i|}\sum_{n=1}^{|\bar{I}_l^j|}avg\left(insCost_I\left(I_k^{i,m},\bar{I}_l^{j,n}\right),delCost_{\bar{I}}\left(\bar{I}_l^{j}\right)\right)-confThr\right)}{\left|I_k^i\right|\cdot\left|\bar{I}_l^j\right|} \tag{5.8}$$

$$insCost_I\left(I_k^{i,m},\bar{I}_l^{j,n}\right)=-log_2\frac{avgconfFTC_I\left(I_k^i,I_k^{i,m},f\right)^2}{\left(avgconfFTC_I\left(I_k^i,\bar{I}_l^{j,n},i\right)\cdot avgconfFTC_I\left(I_k^i,\bar{I}_l^{j,n},f\right)\right)} \tag{5.9}$$

$$delCost_I\left(I_l^{j,n}\right)=-log_2\frac{\left(avgconfFTC_I\left(I_l^j,I_l^{j,n},b\right)\cdot avgconfFTC_I\left(I_l^j,I_l^{j,n},f\right)\right)}{\left(\left(\sum_{y=1}^{|I_l^{j-1}|}avgconfFTC_I\left(I_l^j,I_l^{j-1,y},f\right)\right)\Big/\left|I_l^{j-1}\right|\right)^2} \tag{5.10}$$

Insertion and deletion of activities cannot happen in a random fashion. The inDel scoring enables the insertion of activities concerning (and preserving) a functionality between the underlying activity pairs. On the contrary, the insertion or deletion of activities violating the business conditions is dynamically penalized by *indelScr* component. Figure 5.12 exemplifies an iteration for element (2,3) in the underlying Needleman-Wunsch matrix (F).

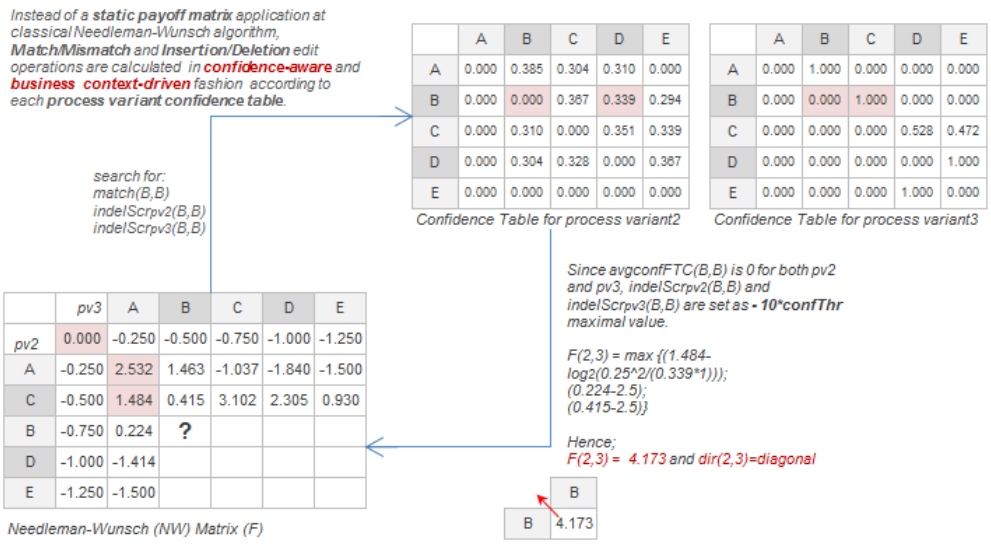


Figure 5.12. A Sample Iteration for Element (2,3) in NW Matrix (F). The major discrepancy from the *classical NW algorithm* given in Figure 5.9 is instead of *static payoff matrix application*, the edit operations are valuated according to *business context* that is implicitly given in the *confidence table* of each process variant.

Backtracking for Constructing the Combined Schema. The basic update and backtracking operations of NW algorithm (i.e. the wave-front concept of NW algorithm [86]) are retained in Multi-Sequence Alignment technique except for the following point: while backtracking throughout the *Needleman-Wunsch matrix* (*F*), the pair of elements is added to indel character array of current elements as follows:

- $S^{i,m}_k$ and $T^{j,n}_l$ if the move is to (*i-1,j-1*). The value change at *simScr*(*S,T*) due to this move is assigned to *structural similarity*. This region of high similarity shared between dominant behaviors is the evidence of common functionality and manifestation of these commonly used

sub-sequences might indicate a functionality inheritance among process variants according to process configuration in BPM [80].

- $S^{i,m}{}_k$ and the gap symbol (–) if the move is to *(i-1,j)* or (–) and $T^{j,n}{}_l$ if the move is to (*i,j-1*). The value change at *simScr*(*S,T*) due to this move is assigned to *behavioral similarity*. Hence discrepancies and extraordinary behaviors are pinpointed at the sub-regions that are intensively filled with gap symbol (−).

After all alignments at the current level are completed, the similarity scores are normalized according to the minimum/maximum (*MIN/MAX*) similarity scores of source individual (*S*). Afterwards, normalized similarity scores of the alignments are summarized at the *similarity matrix* for each level of the process family tree. The alignment with the maximum normalized similarity score is selected as *optimal* and a combined individual (i.e. compound of aligned forms of individuals *S* and *T*) is created and passed to the next level. Alignment and combining, firstly, the most similar individuals, and then gradually adding more distant ones at the following levels continues up to a single combined individual (i.e. *root*) is left.

The major outcome of multi-sequence alignment process is depicted in Figure 5.13 through its application on 5 process variants (*wabo1-wabo5*) for Environmental Permit Application process [11]. Initially, sequence alignment is applied on all pairwise combinations of dominant behaviors of process variants. As seen in the figure, at base level (*level=1*), alignment between *wabo2* and *wabo3* is the least costly (i.e., most similar) one among all pairwise alignments. At level 2, we have 4 entities to align: three individual process dominant behaviors and one combined form of 2 aligned dominant behaviors. At level 5, all process alternatives are aligned and a single combined alignment schema is formed highlighting the common regions and divergences.

Figure 5.13. Sample Process Family Tree for Environmental Permit Application Process. Multi-Sequence Alignment is applied with confidence enhanced scoring functionality and MIN/MAX normalization. *Cutting level* determines the set of clusters, namely *process families,* for the corresponding alignment run.

Pseudo codes for Multi-Sequence Alignment and Alignment operations are given in Algorithm 5.2 and Algorithm 5.3 respectively.

Algorithm 5.2: MultiSequenceAlignment(*process runtime data*, *confidence tables*)

```
1:  REFRESH currIndividualList, currAlignmentRun, currAlignmentRunList
2:  INIT alignRun
3:  while alignRun ≤ maxAlignRun do
4:    INIT currIndividualList FOR alignRun
5:    INSERT currIndividualList INTO individualSet
6:    CREATE currAlignmentRunList FOR alignRun
7:    INSERT currAlignmentRunList INTO alignmentRunSet
8:    INIT levelInd
9:    while levelInd ≤ maxLevel do
10:     currIndividualList ← SELECT individualList FROM individualSet FOR alignRun
11:     INIT srcIndv
12:     while srcIndv ≤ numbIndv(currIndividualList) do
13:       CHECK level of srcIndv ?= levelInd
14:       INIT trgIndv
15:       while trgIndv ≤ numbIndv(currIndividualList) do
```

```
16:           CHECK level of trgIndv ?= levelInd
17:           CREATE currAlignmentRun WITH srcIndv, trgIndv, levelInd
18:           INSERT currAlignmentRun INTO currAlignmentRunList
19:           ALIGN currAlignmentRun
20:           SET characteristics FOR currAlignmentRun
21:           CALCULATE MIN/MAX SimScr FOR currAlignmentRunList
22:         endwhile
23:         NORMALIZE similarity score AT levelInd FOR currAlignmentRunList
24:         optAlignRun ← GET optimal alignment run AT levelInd
25:         INSERT combined individual of optAlignRun TO currIndividualList BY levelInd+1
26:         COPY non-optimal individuals AT currIndividualList BY levelInd+1
27:       endwhile
28:     endwhile
29:     BUILD process family tree FOR alignRun
30:   endwhile
31:   GET the most frequent process family tree FOR ALL alignRun
```

Algorithm 5.3: Align(*currAlignmentRun*)

```
1:   REFRESH NWTable
2:   srcIndv ← srcIndv attribute of currAlignmentRun
3:   trgIndv ← trgIndv attribute of currAlignmentRun
4:   level ← level attribute of currAlignmentRun
5:   INIT i
6:   while i ≤ length(srcIndv) do
7:     INIT j
8:     while j ≤ length(trgIndv) do
9:       INIT diagonalScr, horizontalScr, verticalScr
10:      INIT ns
11:      while ns < numbSeq(srcIndv) do
12:        INIT nt
13:        while nt < numbSeq(trgIndv) do
14:          CALCULATE diagonalScr ← diagonalScr + matchScr(srcIndv^i_ns,trgIndv^i_nt)
15:          CALCULATE horizontalScr ← horizontalScr + indelScr_srcIndv(srcIndv^i_ns,trgIndv^i_nt)
16:          CALCULATE verticalScr ← verticalScr + indelScr_srcIndv(trgIndv^i_nt,srcIndv^i_ns)
17:        endwhile
18:      endwhile
19:      newObj ← CREATE NWObject WITH i, j, diagonalScr, horizontalScr, verticalScr
20:      INSERT newObj INTO NWTable
21:    endwhile
22:  endwhile
```

The complexity of Multi-Sequence Alignment technique is approximately $O(n^3l^2)$, where n stands for the number of candidate process variants ($|PV|$) and l denotes average length of the sequence (or individual).

An Illustrative Example for Multi-Sequence Alignment. In this part, the illustrative example given in Section 5.1.4 is detailed by Multi-Sequence Alignment phase for a sample alignment run, i.e. *run=2*. According to the reference process models and runtime parameter configurations, the *runtime information* (i.e. from-to chart, confidence table and consensus activity sequence coding the dominant behavior per process variant) for 5 distinct example runs is obtained at Dominant Behavior Extraction phase. Figure 5.14 summarizes the runtime information for the underlying alignment run.

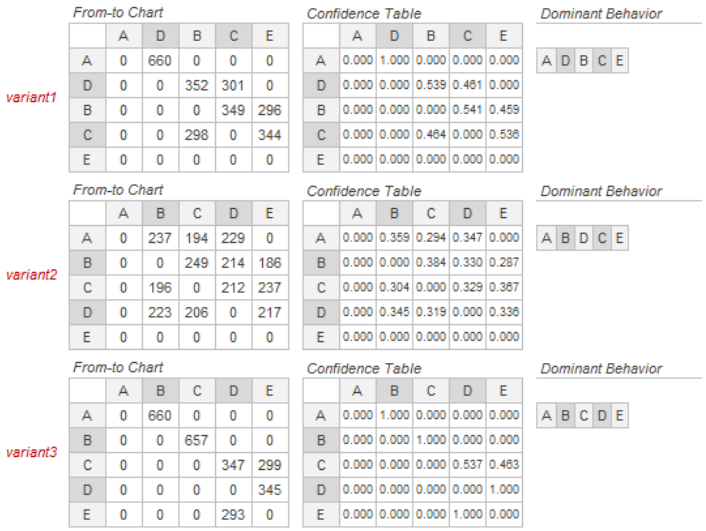


Figure 5.14. Runtime Information per Process Variant (*process variant1–3*) for *run=2*.

At the base level (*level=1*) of Multi-Sequence Alignment, all candidate base-level process variants are aligned on pairwise basis. Then similarity scores obtained through pairwise alignments are summarized as given in Table 5.6. Each similarity score is normalized according to the source individual's (*S*) *MIN/MAX* values. Due to the highest normalized similarity scores (*nSimScr*), *variant2* and *variant3* are selected as the closest individuals and they are combined and transferred as a new individual to the next level as shown in Figure 5.15.

Table 5.6. Similarity Matrix at *level=1.* Similarity scores (*simScr*) are normalized according to source individual *MIN/MAX* values. Hence *alignment*(pv_2,pv_3) is selected and a new individual (i.e. compound of aligned forms of process variants pv_2 and pv_3) is created and passed to the next level (*level=2*).

Similarity Matrix

	pv1	pv2	pv3
pv1	simScr: 17.29 nSimScr: 1.0	simScr: 9.656 nSimScr: 0.0	simScr: 11.918 nSimScr: 0.296
pv2	simScr: 9.656 nSimScr: 0.0	simScr: 10.302 nSimScr: 0.603	simScr: 10.727 nSimScr: 1.0
pv3	simScr: 11.918 nSimScr: 0.118	simScr: 10.727 nSimScr: 0.0	simScr: 20.852 nSimScr: 1.0

	pv3	A	B	C	D	E
pv2	0.000	-0.150	-0.300	-0.450	-0.600	-0.750
A	-0.150	4.146	2.646	1.557	0.691	-0.134
B	-0.300	3.333	8.170	7.453	5.965	5.140
D	-0.450	2.751	7.589	6.871	11.428	10.603
C	-0.600	1.900	6.737	10.869	10.577	10.427
E	-0.750	1.075	5.912	11.061	10.248	10.727

Needleman-Wunsch (NW) Matrix (F)

Backtrack Step	Alignment	Score	Direction
(a)	[] []	10.727	diagonal
(b)	[E] [E]	10.577	vertical
(c)	[CE] [-E]	11.428	diagonal
(d)	[DCE] [D-E]	7.453	horizontal
(e)	[-DCE] [CD-E]	8.17	diagonal
(f)	[B-DCE] [BCD-E]	4.146	diagonal
(g)	[AB-DCE] [ABCD-E]	0	terminated

Backtracking Table

Figure 5.15. Needleman-Wunsch Matrix (F) and Backtracking Table for Alignment *alignment*(pv_2,pv_3) at *level=1*.

At the following and final level, priorly combined individual replaces the process variants *variant2* and *variant3*. Matching and inDel edit operation are valuated according to each inherited variant's confidence tables. Accumulated scores are then normalized by the Cartesian product of source (*variant2*, *variant3)* and target individuals (*variant1*) length. While similarity matrix given in Table 5.7 summarizes the similarity and normalized similarity scores for *level=2*, Needleman-Wunsch matrix (F) and backtracking table of optimal alignment (*alignment*((*pv*$_2$,*pv*$_3$),*pv*$_1$)) are given in Figure 5.16.

Table 5.7. Similarity Matrix at *level=2.*

Similarity Matrix

	pv1	*pv2* *pv3*
pv1	*simScr: 17.29* *nSimScr: 1.0*	*simScr: 10.512* *nSimScr: 0.0*
pv2 *pv3*	*simScr: 10.512* *nSimScr: 0.0*	*simScr: 15.423* *normSimScr: 0.0*

		pv1	A	D	B	C	E
pv2	*pv3*	0.000	-0.150	-0.300	-0.450	-0.600	-0.750
A	A	-0.150	4.885	3.696	2.643	1.793	0.968
B	B	-0.300	3.554	3.643	7.743	6.688	5.863
-	C	-0.450	2.338	2.881	6.527	11.217	10.900
D	D	-0.600	1.176	3.955	5.365	10.054	10.413
C	-	-0.750	0.750	3.984	4.939	9.629	10.662
E	E	-0.900	-0.075	3.159	4.114	8.804	10.512

Needleman-Wunsch (NW) Matrix (F)

Backtrack Step	*Alignment*	*Score*	*Direction*
(a)	[] [] []	10.512	*vertical*
(b)	[E] [E] [-]	10.662	*vertical*
(c)	[CE] [-E] [--]	10.413	*vertical*
(d)	[DCE] [D-E] [---]	10.9	*horizontal*
(e)	[-DCE] [-D-E] [E---]	11.217	*diagonal*
(f)	[--DCE] [C-D-E] [CE---]	7.743	*diagonal*
(g)	[B--DCE] [BC-D-E] [BCE---]	3.696	*horizontal*
(h)	[-B--DCE] [-BC-D-E] [DBCE---]	4.885	*diagonal*
(i)	[A-B--DCE] [A-BC-D-E] [ADBCE---]	0	*terminated*

Backtracking Table

Figure 5.16. Needleman-Wunsch (F) Matrix and Backtracking Table for Alignment *alignment((pv₂,pv₃),pv₁)* at *level=2*.

Consequently as shown in Figure 5.17, *process family tree*, a dendrogram-like output displaying the grouping of process alternatives, similarities and differences is created for *run=2*. The cluster contents at a predefined *cutting-level* are referred to as *process families*. If a predefined cutting level is considered on this tree, *variant2* and *variant3* are composing a process cluster, i.e. *cluster1* for *run=2*. Additionally, overlapping region (i.e. AB) emphasizes a functional inheritance for the underlying process variants, i.e. *variant2* and *variant3*.

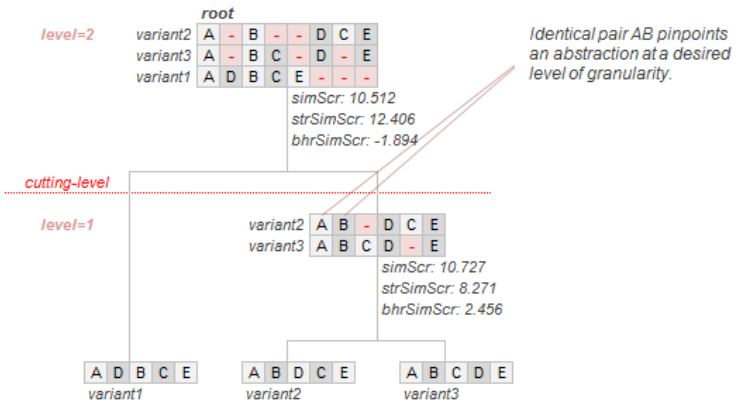


Figure 5.17. Process Family Tree for Sample Run for *run=2*. While *cluster1* is composed of process variants *variant2* and *variant3*, *variant1* is assigned to *cluster2*.

The results of the Multi-Sequence Alignment phase is stored as an alignment matrix, and overlapping regions are analyzed by identical pairs feature sets for process configuration as explained in Section 5.3. The underlying multi-sequence alignment phase is implemented at prototype software named Confidence Enhance Multi-Sequence Alignment (ConfEnhMSA). Details about the functionality and interface design are given in Appendix A.

5.2.3. Process Family Construction by Pairwise Alignment

Pairwise Alignment technique is an adaptation of NW algorithm and also a former version of Multi-Sequence Alignment, which exploits the similarity scores between process variants (*pv*) on a pairwise alignment basis. While confidence-enhanced cost functioning constitutes the baseline for Pairwise Alignment and all equations in Multi-Sequence Alignment are valid for this NW adaptation, this technique is performed at *only base-level* (*level=1*).

Due to the progressive alignment nature of Multi-Sequence Alignment, the process families, which are the clusters grouping the process variants (i.e. organizations), are built upon by dendrogram formation. As an alternative approach to form process families, we can group the process variants on the basis of the pairwise alignment scores. For this aim, we apply a *data preprocessing step* that transforms the similarity scores into distance attributes. As the similarity scores obtained by Pairwise Alignment are highly-correlated to the confidence tables of both source and target process variant, these scores should be normalized and this normalization should hold the following properties:

i. $dist(pv_0, pv_1) \geq 0$, *all* $pv_0, pv_1 \in PV$, *the set of all process models (non-negativity property)*
ii. $dist(pv_0, pv_1) = dist(pv_1, pv_0)$ *all* $pv_0, pv_1 \in PV$ *(symmetry property)*
iii. $dist(pv_0, pv_1) = 0$, $pv_0 \equiv pv_1$
iv. $dist(pv_0, pv_1) + dist(pv_1, pv_2) \geq dist(pv_0, pv_2)$ *(triangle inequality property)*

According to these properties, *cosine similarity* is applied as a way to normalize the similarity scores (*simScr)* and convert these values to [-1, 1] value range. Prior to cosine similarity calculation, *process variant vector* concept is defined as follows:

Definition **(process variant vector, pv_i^k).** Let pv_i^k be a vector of (source) process variant *i* at alignment run *k* holding similarity scores (*simScr*) with each of the process variants *j*. The term weight *j* (pv_{ij}^k) is the similarity score between process variant *i* and *j* at alignment run *k.*

Definition **(cosine similarity, $cosSim_k(pv_i,pv_j)$).** Let pv_i^k and pv_j^k be the corresponding similarity vectors for process variants *i* and *j* at alignment run *k* respectively. As stated in [91], *cosine similarity* between variants *i* and *j* denoted by $cosSim_k(pv_i,pv_j)$ is the cosine of the angle between those similarity vectors, given in Equation 5.11. The value of cosine similarity ranges from -1 (quite distinct) to +1 (equivalent).

$$cosSim_k\left(\vec{pv}_i, \vec{pv}_j\right) = \frac{\vec{pv}_i \times \vec{pv}_j}{\left|\vec{pv}_i\right| \cdot \left|\vec{pv}_j\right|} = \frac{\sum_{l=1}^{N} \vec{pv}_{il} \cdot \vec{pv}_{jl}}{\sqrt{\sum_{l=1}^{N} \vec{pv}_{il}^{\,2}} \cdot \sqrt{\sum_{l=1}^{N} \vec{pv}_{jl}^{\,2}}} \tag{5.11}$$

Figure 5.18 demonstrates the normalization of similarity scores (*simScr*) through cosine similarity (*cosSim*) for the illustrative example. Figures 5.19 and 5.20 depict similarity scores and cosine similarity values per alignment run for *variant1* and *variant3* as the source process variants (alternatives).

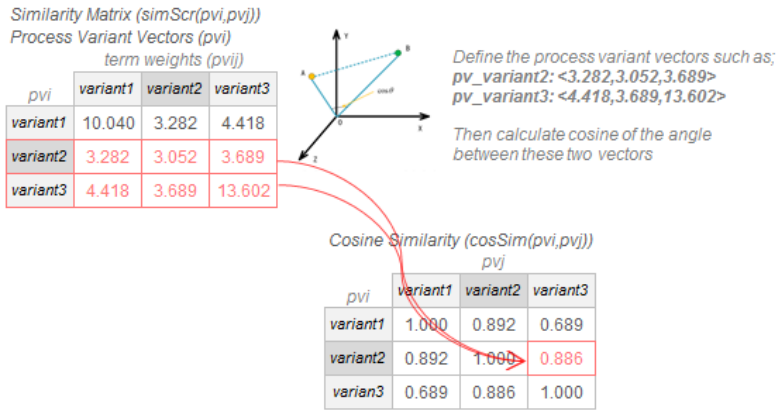


Figure 5.18. Preprocessing Step for Pairwise Alignment. Similarity scores (*simScr*) obtained are normalized into cosine similarity values (*cosSim*) for sample *run=1*.

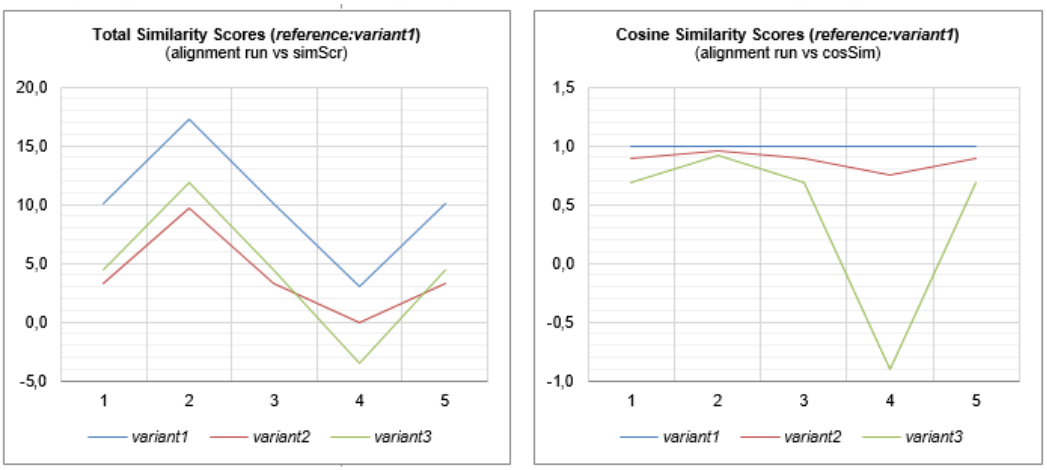


Figure 5.19. Similarity (*simScr*) and Cosine Similarity (*cosSim*) Values for *variant1* as Source Process Variant (*X-axis:alignment runID, Y-axis:similarity score*). The [0.8, 1.0] cosine similarity range highlights a significant commonality among *variant1* and *variant2*.

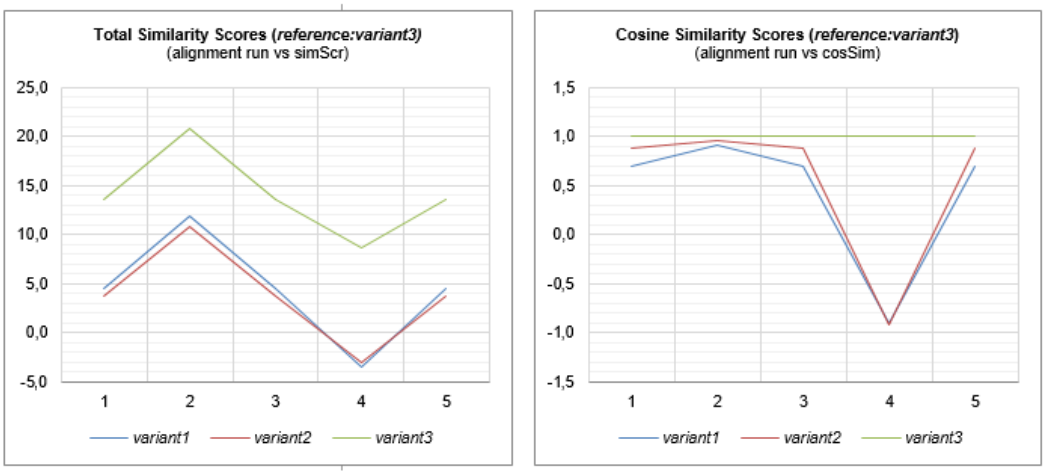


Figure 5.20. Similarity (*simScr*) and Cosine Similarity (*cosSim*) Values for *variant3* as Source Process Variant (*X-axis:alignment runID, Y-axis:similarity score*). While cosine similarity between *variant2* and *variant3* pinpoints a significant commonality, there happens a distinction between the corresponding variants at *run=4*.

Prior to *process clustering*, cosine similarity values (*cosSim*) are transformed into distance attributes (*dist*). This transformation is applied according to $\mathrm{dist}(\mathrm{pv_i},\mathrm{pv_j})=\sqrt{1\text{-}\mathrm{cosSim}(\mathrm{pv_i},\mathrm{pv_j})^2}$ formula. In this aspect, *dist(pv$_i$,pv$_j$)* attribute refers to the distance between source process variant *i* (*pv$_i$*) and candidate process variant *j* (*pv$_j$*). Additionally, each transformed instance (or observation) has a nominal attribute, namely *target class*, indicating relevant source process variant. This attribute is important to interpret the content of the process families (clusters) and to create the confusion matrix. Figure 5.21 demonstrates the transformation from cosine similarity (*cosSim(pv$_i$,pv$_j$)*) to distance attribute (*dist(pv$_i$,pv$_j$)*).

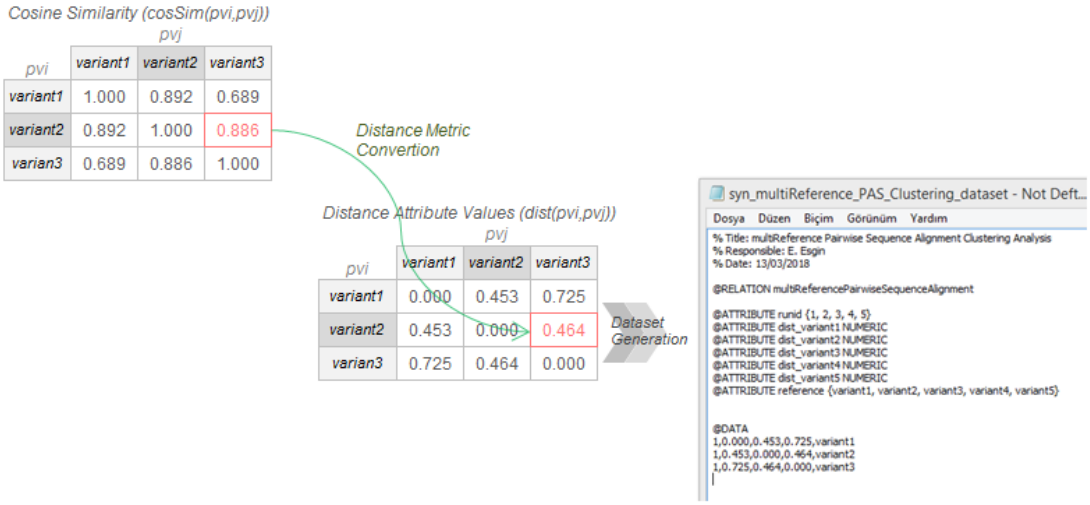


Figure 5.21. Preprocessing Step for Pairwise Alignment. Cosine similarity values are converted into distance attributes. Hence each line for the underlying alignment run implies to a distinct instance (or observation). *Reference* nominal attribute is used to interpret the content of the clusters.

Finally, we apply various clustering algorithms (e.g. K-Means or agglomerative hierarchical clustering (AHC)[6]) to partition the process variants (or organizations) into process families, where predictor variables are related to the distance attribute values, i.e. *dist(pv$_i$,pv$_j$)*. As stated in [34, 41, 62], agglomerative hierarchical clustering is widely applied at trace clustering application in process mining domain. Figure 5.22 shows the instance plot, which visualizes each distinct instance according to the distance attributes (*dist_varianti*), with 15 instances for the sample case (i.e. 5 alignment runs × 3 process alternatives). Although, the distance values are staggered at [0.25, 0.5] interval, especially the distance values at *run=4* may result in a process clustering between *variant2* and *variant3.*

[6] Clustering algorithms are applied with MEAN criteria (i.e. the mean distance of a merged cluster) and Eucledian distance function at Weka 3.8 software.

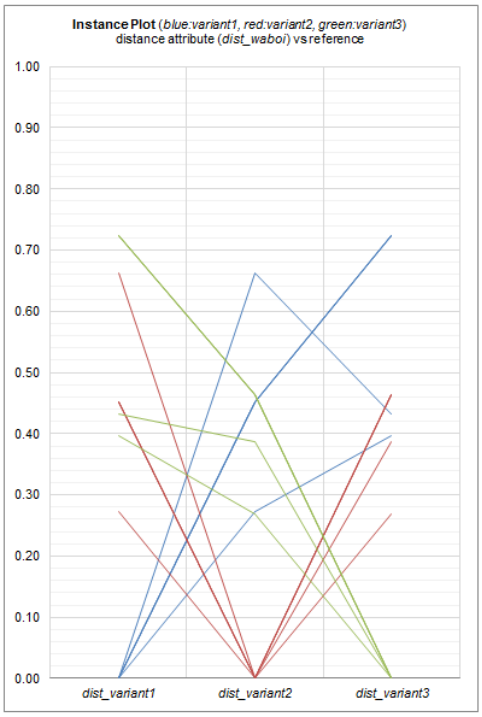


Figure 5.22. Instance Plot for Sample Case (*X-axis:distance attributeID, Y-axis:distance value*). Due to the overlapping instances, the number of instance lines is lessened and the color of these lines is turned into darker color.

5.3. Process Configuration

Complying with the software as-a service (SaaS) paradigm, which refers to a software distribution model that hosts the applications or solution services at a distributed system by a vendor or service provider and they made available these products to customers over a network upon request [12], the recordings of event logs can be provided in a unified manner across the organizations. Moreover process variants and configurations can be managed and related to the process enactment. The monolithic view of traditional process mining is evolving through cross-organizational process mining to analyze and support these multi-tenant processes. The basic idea of process configuration is to generate a model that does not address the corresponding process alternatives to a sole process but unifies them into a family of processes [13]. Hence the configured model has fever number of activities due to the removal of potential behaviors during configuration. While collaboration setting aims the interoperability among different organizations by distributing the processes over different process observers, exploiting the commonality aims to share the process knowledge and deal with the variability among the organizations.

In this aspect, *process configuration* phase aims to explore common patterns of activity invocations at organizations within the same process family. These common patterns are defined by two feature sets: *identical pairs* (IP) and *maximal identical pairs* (maxIP). Afterwards these structures are visualized by alignment matrix on alignment run basis.

5.3.1. Feature Sets Derivations

Similar regions (i.e. sequence of activities) that are common across a set of traces in event logs emphasize some sort of common functionality for the underlying process. Alternatively, a region that is pinpointed by high similarity measurements between multiple sequences can be a proof of functionality share among the corresponding process candidates [33, 34]. Indeed, deriving these overlapping regions by feature sets enable the clustering of process alternatives such that, process alternatives sharing relatively frequent and longer covered regions enforce these alternatives to be assigned to the same process family. We formally define these common process constructs in terms of two feature sets:

***Definition* (identical pairs, *IP*).** An *identical pair* (*IP*) in a multi-sequence or pairwise alignment run, A_i, is a pair of matching subsequences *x* and *y* such that, the activity to the immediate neighbor of subsequence *x* is different from the symbol to the immediate neighbor of subsequence *y*. This feature set is similar to maximal pair in [58].

***Definition* (maximal identical pairs, *maxIP*).** A maximal pair (*maxIP*) is defined as an *IP* that is never subsumed as a substring of any other *IPs* at *any alignment run* and the length of underlying *IP* should be greater than 1-unit.

Table 5.8 exemplifies the derivation of maximal identical pairs feature sets of the process family for the illustrative example. Derived maximal identical pairs are characterized by three attributes: *Order* refers to the length of the maximal identical pair and *frequency* is the occurrence rate throughout all alignment runs. *Coverage* holds the span of the subsequence with respect to the total length of alignment.

Table 5.8. Derivation of *IP* and *maxIP* Feature Sets at Process Families. *IPs* in *grey-shaded* regions (AB, D and E) in Figure 5.17 pinpoint high similarity shared between sequences.

Process Family	*Identical Pairs (IP)*	*IP Describtion*	*Order*	*Frequency*	*Coverage*	*maxIP*
variant2 variant3	{AB}	*{activity_A, activity_B}*	2	1.000	0.333	*yes*
	{D}	*{activity_D}*	1	1.000	0.167	*no*
	{E}	*{activity_E}*	1	1.000	0.167	*no*

5.3.2. Alignment Visualization

Process mining employs various visualization methods varying from presentation of overview results to presenting directed insights, merging different analysis directions. However, most of the current visualization methods in process mining fall short when dealing with large datasets of event logs. Visualization is also intensively used in different areas within bioinformatics [23]. Especially, sequence exploration and visualization techniques can be adapted to analyze process similarity measurement outcomes [11, 27]. Respectively, we design the *alignment matrix*, which decomposes the aligned elements (e.g. the gap symbol or activity labels) on the alignment run basis. This view of the alignment matrix provides a holistic insight, in which discrepancies and extraordinary activity invocations are viewed in regions that are pinpointed with the gap symbol (–). Additionally, concurrent activity invocations indicated by maxIP may manifest a shared business context among process alternatives.

Table 5.9 shows the alignment matrix summarizing all alignment runs, at which process alternatives *variant2* and *variant3* are both assigned to the same process family, i.e. alignment run 1, 2, 3 and 5. While gap symbol (–) is highlighted in red color, identical pairs are grey-shaded. Correspondingly, the maxIP, i.e. AB (with average 1.0 frequency and 0.33 coverage values) emphasizes a process semantics commonality for the underlying process variants.

Table 5.9. Alignment Matrix for {*variant2, variant3*} Process Family. Average coverage refers to total span of IPs at the corresponding alignment run.

runID	*process alternative*	1	2	3	4	5	6	*average coverage*
1	*variant2*	A	B	D	C	-	E	0.667
	variant3	A	B	-	C	D	E	
2	*variant2*	A	B	-	D	C	E	0.667
	variant3	A	B	C	D	-	E	
3	*variant2*	A	B	D	C	-	E	0.667
	variant3	A	B	-	C	D	E	
5	*variant2*	A	B	D	C	-	E	0.667
	variant3	A	B	-	C	D	E	

Accordingly, it is aimed to depict and explore this expression of commonly invoked sub-processes among similar process alternatives in the same process family. These conserved regions can be interpreted as a *functional inheritance* at enactments of process alternatives and we can adapt these regions as a mean to form *abstractions* at configurable process models as shown in Section 6.5.2.

CHAPTER 6

EXPERIMENTAL ANALYSIS

6.1. Overview for Experimental Analysis

In this chapter, dominant behavior extraction, sequence alignment and process configuration phases are briefly evaluated with respect to the *use cases*, which are *Travel Management, Loan Application, Environmental Permit Application* and *Period-End Closing*. These real-life use cases can be characterized by *data source* and *data type* attributes as follows:

a. *Data source*. The *problem domain* is the universe that covers the as-is and to-be process models, the organizations, process observers (or domain experts), standard operation procedures and the business rules. The data source refers to the source from which event logs or reference process models are gathered. These sources are process-aware information systems as well as process mining knowledge base or repositories (e.g. APROMORE[7]).
b. *Data type*. In process-aware information systems, the event logs are staggered around the transactional database tables and it may be infeasible to assign the transactions (or events executions) to the process instances (cycles). This issue is called *unlabeled event log* rationale in the literature [47, 87]. Due to the difficulty of collecting real-life event log data from information systems, we developed a program to generate the event logs according to reference process structures and Petri net's firing rule. Details about synthetic event log generation are given in Section 6.2.2.
   In the case of process mining knowledge base, the benchmark event log is converted from XES (IEEE Standard for eXtensible Event Stream) standard to the custom text file format that is used in the process discovery algorithm implementation.

The experiments are designed to evaluate the phases of the corresponding framework. As stated in Chapter 5, proposed approach is composed of three phases: dominant behavior extraction, sequence alignment and process configuration.

- *Dominant behavior extraction*. In the context of generalization, simplicity, precision and fitness quality dimensions emphasized in process discovery [20, 23, 53], dominant behavior extraction performance is evaluated in terms of completeness and soundness metrics. While, the completeness is similar to fitness in [23, 89] and recall in [19, 50], the soundness resembles minimality or behavioral appropriateness in [89] and precision in [19, 23, 50]. In addition to quality metrics, we analyze the performance of Genetic Algorithms (GA) engine applied at dominant behavior extraction. Hence various data visualization and statistical tests are applied to analyze the effect of GA drivers (i.e. schema application, population size and probability of crossover) at population convergence.

- *Sequence alignment*. This core phase includes two major approaches: Pairwise and Multi-Sequence Alignment and Pairwise Alignment within two distinct settings: *single-reference* and

[7] APROMORE (Advanced Process Analytics Platform) is an open-source business analytics platform that combines current process mining approaches with the functionalities of process model repositories (www.apromore.org).

multi-reference. At Single-Reference Pairwise Alignment setting, a unique and predefined process alternative is selected as the *reference* and all sequence alignment runs are applied for this single reference-candidate combination. On the other hand, each process alternative is selected as reference for once at multi-reference setting and all reference-candidate combinations are handled at alignment runs. In the case of Single-Reference Pairwise Alignment application, we aim to measure the performance of the underlying approach to reflect the perception of process observers in process similarity measurement. Therefore we collect the intuitive judgments in process similarity measurements by a questionnaire that includes the process maps of reference and all candidate process alternatives. We analyze these ordinal similarity rankings at *likert charts* and then various information retrieval (IR) metrics (i.e. *cosine similarity*, *discount cumulative gain*) and *recall/precision framework* adapted from [24] are applied to measure the correlations between the results of proposed approaches and intuitive judgments. Additionally, we introduce *semantic similarity* metric, which is based on identical pairs (IP) at sequence alignments and the likelihood between distance function concepts in [57]. This metric interprets the fundamental mechanism that determines the alignment context (i.e. matching or inDel edit operations) of the dominant behavior and cost function according to the process structures. We analyze the performance of Pairwise Alignment in comparison to the former version of the approach in the literature, namely standard NW-Needleman Wunsch and CANW-Confidence-aware Needleman Wunsch approaches introduced in [66, 67].

Multi-Reference Pairwise Alignment is able to return normalized similarity scores and these scores can be converted to distance attribute. Hence various clustering algorithms are applied with these distance values and the content of derived *process families* is compared with prior studies in literature handling the same use case. Due to the progressive alignment fashion of Multi-Sequence Alignment, it is also possible to analyze the topology and the branching orders of the *process family tree* at different cutting levels. Likewise in Multi-Reference Pairwise Alignment, the content of process families (clusters) is compared with prior studies in the literature. Additionally, the clustering quality of the alignment modes are interpreted according to intra-cluster distance, inter-cluster distance and silhouette measure metrics.

- *Process configuration.* This phase aims to explore common (conserved) patterns or deviations at dominant behavior alignments. These patterns are the basis for deriving abstractions and process encapsulations at configurable process models and they are refined by *identical pairs* (IP) and *maximal identical pairs* (maxIP) feature sets. Then, it is aimed to manually design the configurable process models according to these common or deviated regions.

The overview for experimental analysis phase is depicted in Figure 6.1.

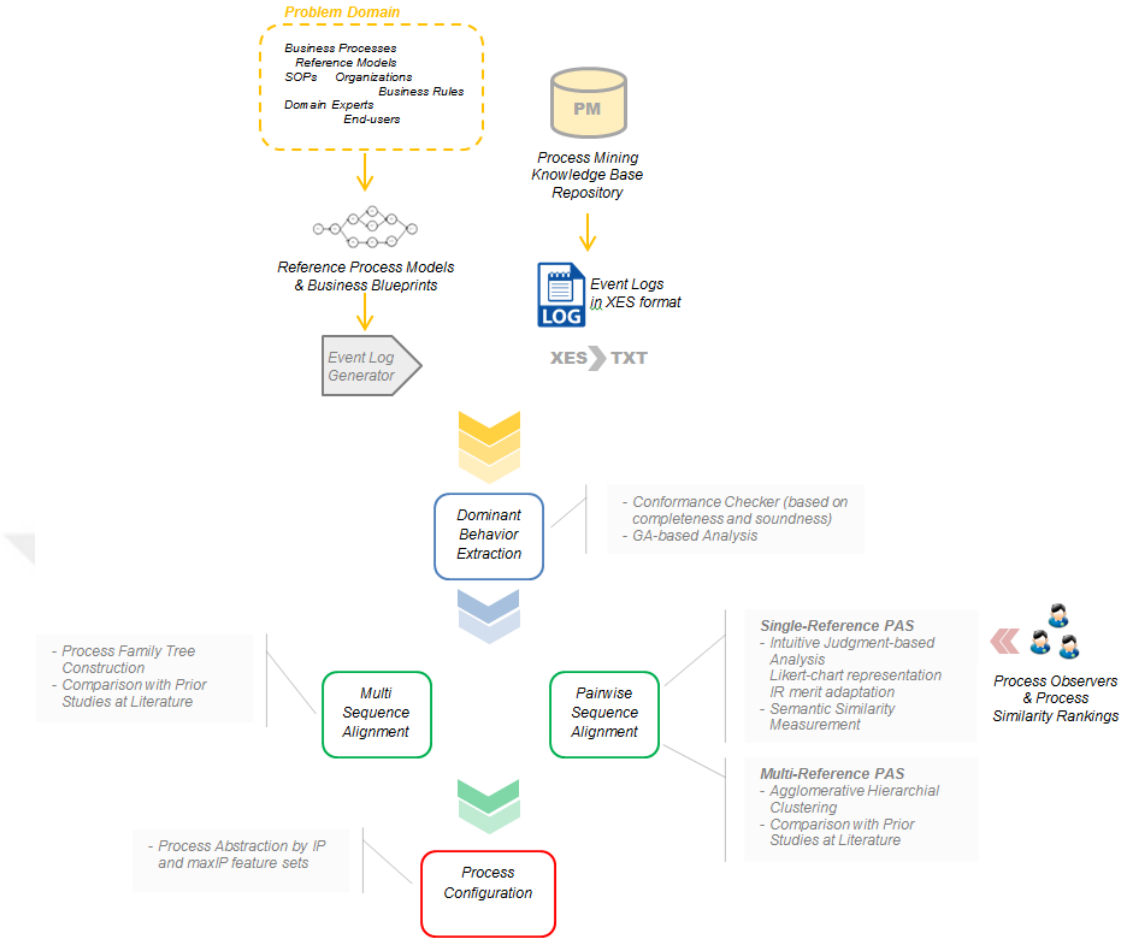


Figure 6.1. Overview for Experimental Analysis.

6.2. Use Cases

The use cases that are handled in the context of experimental analysis are evaluated with respect to problem and solution domain aspects as follows in Table 6.1. While Loan Application and Environmental Permit Application are benchmark use cases that are referenced in process mining literature, Travel Management and Period End Closing use cases depend on the author's SAP experience. This fact affected the source system and data set type features of the underlying use cases as given in Table 6.1.

Although tacit process variant assumption states the fact that there may be more than one process variant in a single event log, it is assumed there exists a single valid process variant per organizations (or process candidate).

Table 6.1. Use Cases with respect to Problem and Solution Domain Aspect.

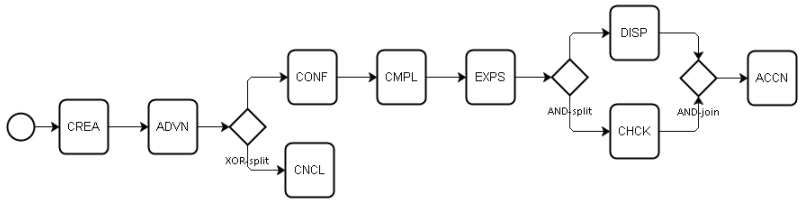
Problem Domain				*Solution Domain*					
Use Cases	Alias	Source System	Data Set Type	**Dominant Behavior Extraction**		**Sequence Alignment**			**Process Configuration**
				Process Discovery Based Analysis	*Genetic Algorithms Based Analysis*	*Pairwise Alignment*	*Multi-Sequence Alignment*	*Intuitive Judgment Based Analysis*	*Comparison with Prior Approaches*	
Travel Management	TRV	SAP/ERP	Synthetic	*Conformance checking (wrt completeness and soundness)*	*Runtime visualization and statical analysis for schema application and croosover operation.*	*Single-Reference PAS*		*Likert-chart representation IR merit adaptation*	*Semantic similarity measurement*	
Loan Application	LA	PM Repository	Benchmark	*Conformance checking (wrt completeness and soundness)*	*Runtime visualization and statical analysis for population size.*	*Single-Reference PAS*		*Likert-chart representation IR merit adaptation*	*Semantic similarity measurement*	
Environmental A	WABO	PM Repository	Benchmark	*Conformance checking (wrt completeness and soundness)*		*Multi-Reference PA and comparison with prior studies at literature*	*Process family tree construction and comparison with prior studies at literature*			*Process abstraction by feature sets*
Period End Closing	PEC	SAP/ERP	Real Life	*Conformance checking (wrt completeness and soundness)*		*Multi-Reference PAS*	*Process family tree construction*			*Process abstraction by feature sets*

6.2.1. Process Description and Event Log Dataset

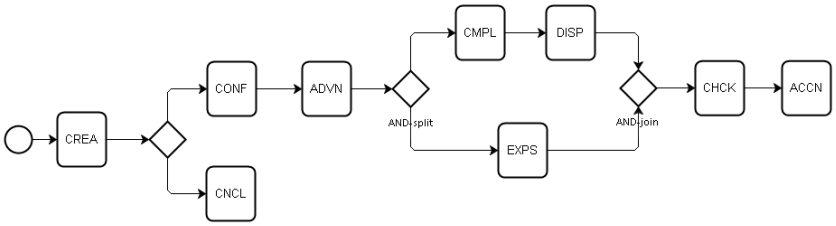
Travel Management business process (alias: *TRV*) is a hybrid business process that is both executed at SAP/FI (SAP Financial Management) and SAP/HR (SAP Human Resources) modules. In this aspect, personnel relevant data (e.g. personnelID, travel request, personnel level and grade) and detailed information about the corresponding travel request (e.g. travel request creation or cancelation, the travel duration and destination) are managed at SAP/HR module, whereas finance relevant transactions (e.g. advance payment term/conditions and expense payment) are managed at SAP/FI module. Respectively, travel management business process aims to manage all travel-specific transactions including trip booking and expense accounting related to the corresponding travel request.

The source information system for the corresponding business process is SAP[8]. SAP provides a wide range of reference model repository, which is expressed in terms of Event-driven Process Chain (EPC) diagram. This knowledge base aims to hold the best-practices to describe the stakeholders to customize, implement and use the ERP system in a more efficient manner. These reference business models are dedicated to different industries which are composed of manufacturing, telecommunication, service and software development [26]. In several ERP implementations, these process models are directly referred as *business blueprint*. This documentation is a composition of software requirement specification (SRS) and software design document (SDD) and it is also used as a contract between the client and the ERP vendor.

Indeed, the process enactment may be quite different from the reference process model; the system conceptualization in terms of SRS and SDD documentations may be inconsistent with the business requirements or the process observers may seek process workarounds instead of standardized know-how's. As the starting point, we handle the reference process models, which are proposed in the business blueprint document for 6 distinct SAP project implementations. While one of these process models is evaluated as *reference*, the latter models are categorized as *candidate* as shown in Figure 6.2. The *activity vocabulary* (i.e. the value range that holds all valid activity labels or transaction codes in SAP) is composed of nine activities as follows; CREA-Travel request create, DISP-Travel request display, ADVN-Advance payment, CONF-Travel request confirmation, CNCL-Travel request canceled, CMPL-Travel completed, EXPS-Expense payment, ACCN-Transfer to accounting and CHCK-Last check. Table 6.2 summarizes process execution characteristics per each process alternatives.

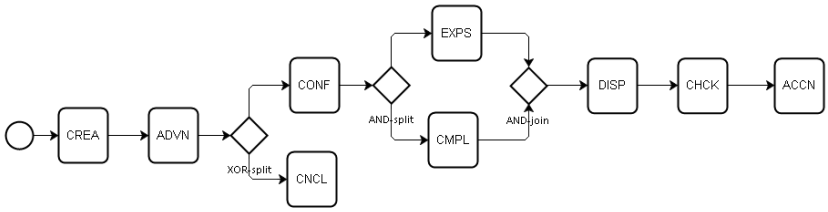


Reference Process Model

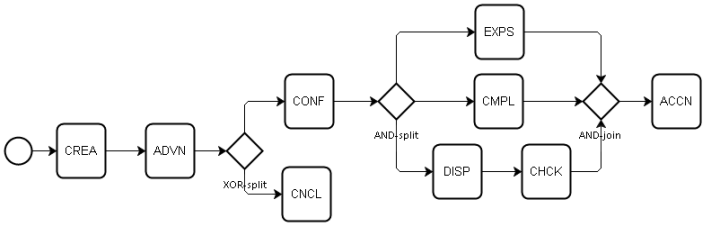


Candidate1 Process Model

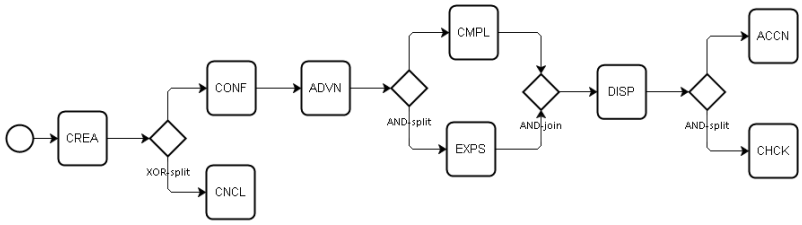
[8] SAP is the abbreviation of "Systems, Applications and Product" for German Enterprise Resource Planning (ERP) System vendor.



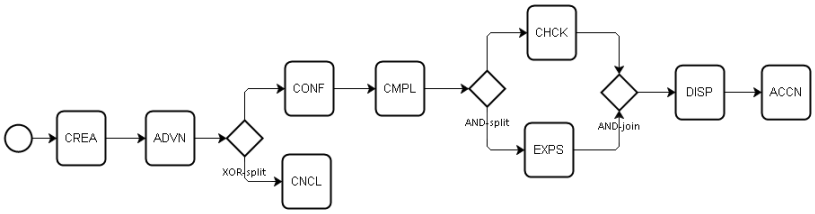
Candidate2 Process Model



Candidate3 Process Model



Candidate4 Process Model



Candidate5 Process Model

Figure 6.2. Proposed Process Models for Reference (*reference*) and Candidate Process Alternatives (*candi*) for Travel Management Use Case.

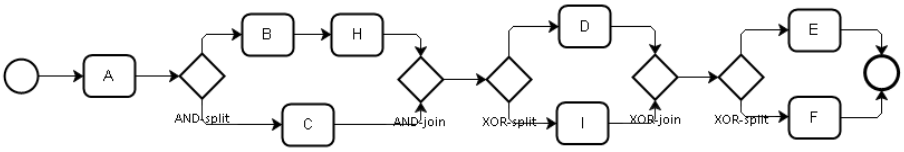
250 synthetic process instances are generated per process alternative by event log generator introduced in Section 6.2.2.

Table 6.2. Process Execution Characteristics for Travel Management Use Case.

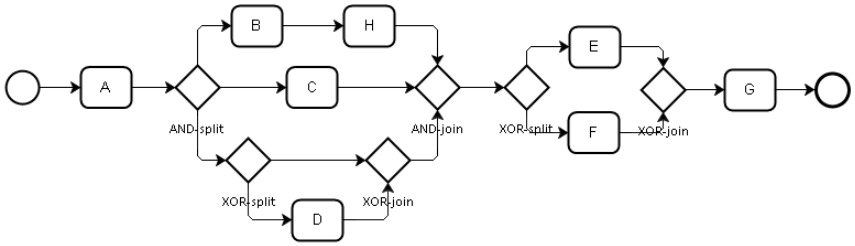
Process Alternative	*Number of Traces*	*Number of Events*	*Number of Activities*
reference	250	1811	9
cand1	250	1817	9
cand2	250	1839	9
cand3	250	1792	9
cand4	250	1830	9
cand5	250	1821	9

The second example is a real-life use case namely Loan Application (alias: *LA*) of a financial institute, providing small consumer credit through a webpage [15, 27]. While reference process model (denoted by *reference*) depicts idealized process imposed in the business blueprint, all four process alternatives (denoted by *candi*) describe the process alternatives for handling loan applications. Even though the processes slightly differ, each process is initiated by sending an e-mail (activity A) and in the end either accepts (activity E) or rejects (activity F) the application [15, 27].

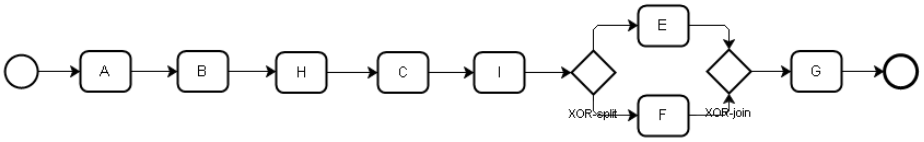
Activity vocabulary consists of 9 activities as follows: A–send e-mail to applicant, B–send check credit request, C–calculate capacity, D–check system, E–accept, F–reject, G–send e-mail, H–process check credit request response and I–check paper archive. The reference process model and process variants are given in Figure 6.3.



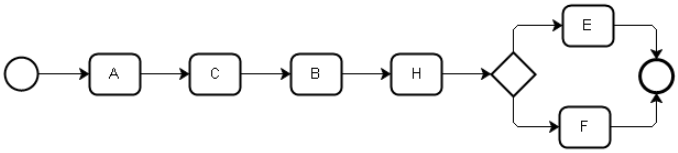
Reference Process Model



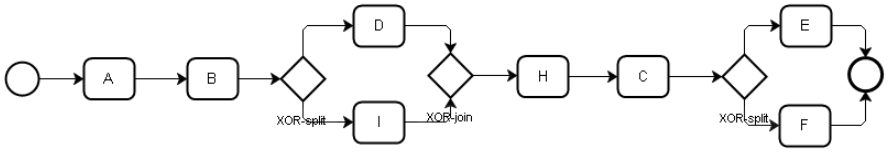
Candidate1 Process Model



Candidate2 Process Model



Candidate3 Process Model



Candidate4 Process Model

Figure 6.3. Proposed Process Models for Reference (*reference*) and Candidate Process Alternatives (*candi*) for Loan Application Use Case.

The event logs for Loan Application use case are obtained from process mining repository and then converted from XES format to the internal text file format that is valid for the corresponding process discovery application [88]. Table 6.3 summarizes process execution characteristics for each process alternatives.

Table 6.3. Process Execution Characteristics for Loan Application Use Case.

Process Alternative	*Number of Traces*	*Number of Events*	*Number of Activities*
reference	250	1471	8
cand1	250	1529	8
cand2	250	1701	8
cand3	250	1222	6
cand4	250	1456	8

Environmental Permit Application (*alias:WABO*) business process in *Configurable Services for Local Governments (CoSeLoG)* project, which investigates the similarities and deviations between processes of different municipalities in Netherlands, aims to handle the building permits process [15, 27]. Five municipalities from CoSeLoG project are collaborating on the underlying business process and jointly selected and configured a shared information system, i.e. Shared Business Process Management Infrastructure (SBPMI), to support this process. The long-term goal of the municipalities is to centralize and standardize the process to reduce the operational costs [15, 27]. Therefore, it is beneficial for the municipalities to share their proven best practices, to understand individual discrepancies between these process alternatives and pinpoint the commonalities among them. This gradual progress can be feasible by using cross-organizational process similarity measurement.

The event log dataset contains records of receiving phase for the building permit application process in 5 municipalities (i.e. *waboi*). The corresponding process alternatives are analogous since the corresponding activity vocabulary is unified for all the municipalities [15, 27]. In this dataset [90], there are 1214 process instances, 2142 events and 27 activities as the lump sum. Table 6.4 summarizes process execution characteristics per each process alternatives and activity vocabulary is given in Table 6.5.

Table 6.4. Process Execution Characteristics for Environmental Permit Application Use Case.

Process Alternative	*Number of Traces*	*Number of Events*	*Number of Activities*
wabo1	54	131	15
wabo2	302	586	13
wabo3	37	73	9
wabo4	340	507	9
wabo5	481	845	23
total	1214	2142	69

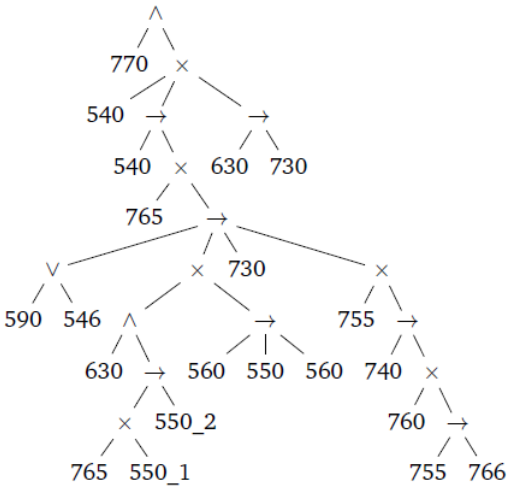
Table 6.5. Activity Vocabulary and Activity Label Mappings (*CoSeLog activityID:activityID*) for Environmental Permit Application Use Case.

CoSeLoG ActivityID	Description	ActivityID
540	Objection to disposal submitted	B
546		%
550	Treat objection	C
560	Objection wrapped up	F
590	Received request for preliminary verdict	G
600	Treat preliminary verdict	H
610	Preliminary verdict wrapped up	I
630	Appeal set	J
640	Received request for preliminary verdict	K
670	Treat appeal	L

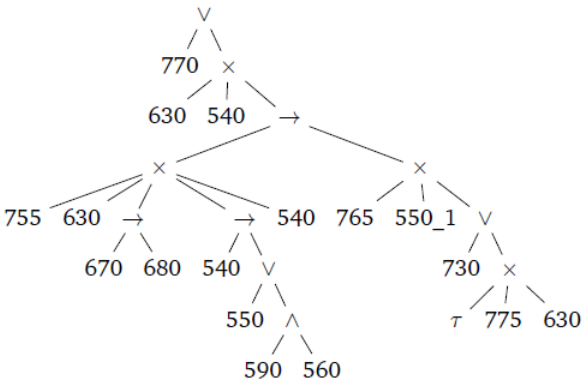
680	Appeal wrapped up	M
700	Higher objection started	N
730	Contested disposal affected	O
740	Verdict given by court	Q
755		}
760	New decision or new evaluation	P
765	Phase start 2	R
766	New decision or new evaluation	S
770	Establish decision phase original decree	T
775	Decision phase definite	U
790	Establish decision phase of the verdict of court	Y
550_1	Treat objection subcase	D
550_2	Treat objection subcase finished	E
650_1		?
650_2		+
780_1	Create decree for the purpose of the disposal of the court	V
780_2	Connect disposal court	W
780_3	Register date of disposal of court	X
STRT	Start	A
FNSH	End	Z

Figure 6.4 depicts the reference process models per process alternative in the form of *process maps*, which denotes the business processes in different business process modeling notations such that;

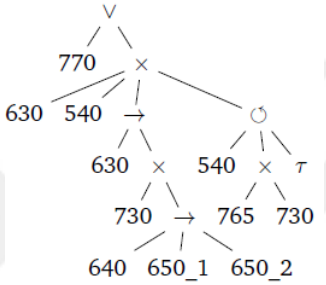
- Process maps are shown in terms of (process) tree and process is triggered at the root node.
- While ∧ stands for an AND-type gateway, ∨ symbol refers to an OR-type gateway and × is used as XOR-type gateway. All process alternatives are valid in terms of gateway type.
- → symbol denotes a direct-successor transition among the left and right-hand side of the process tree. This notation refers to an *in-order traversal* at process trees such that, at first the left sub-process is decomposed (or traversed) and then right sub-process follows this decomposition.
- τ symbol refers to a process termination.



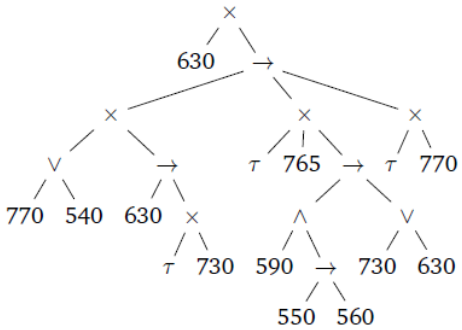
wabo1 Process Map



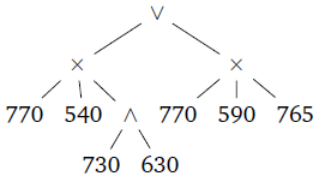
wabo2 Process Map



wabo3 Process Map



wabo4 Process Map



wabo5 Process Map

Figure 6.4. Proposed Process Maps for Process Alternatives (*waboi*) for Environmental Permit Application Use Case.

As the last use case, *Period-End Closing* is a real-life business process, which is managed at SAP/CO (Controlling) module. This business process majorly aims to distribute the overhead costs among cost centers, calculate the activity unit price for each work centers and finalize unit manufacturing costs for semi-finished and finished products. As a result, periodically incurred fixed and variable costs are transferred to major outcomes (i.e. finished products and services) of the organization, the variance between plan and actual costs is analyzed and the cost of goods sold (COGS) is revaluated in an organizational profitability perspective.

Due to the industry standards and business requirements, different *SAP components* (or sub-modules) are configured throughout the customization phase. For instance, while Cost Center Accounting (CCA) component is crucial for both manufacturing and service industry, especially Product Costing (PC) is solely valid for the organizations in manufacturing industry. According to author's SAP consultancy experience, 5 organizations (namely *clients*) with distinct SAP/CO component configurations are determined within scope of this case study. As stated in [27, 28], it is sometimes infeasible to retrieve event-log data in the form of <*caseID, activityID*>. Since the vast array of events is staggered around various application tables at the process-aware information system and it becomes difficult to correlate the events to specific process instances [27, 28].

SAP transactions related to Period-End Closing business process are collected from SAP systems via ST03-Workload Monitor transaction and each transaction is mapped to an activityID. Hence 400 process instances for each organization (i.e. each financial period×plant cartesian refers to a distinct process instance) are retrieved from source SAP system. Table 6.6 gives the details and characteristics of the underlying organizations and active SAP/CO component set. Basically, *client2* and *client4* are evaluated as service industry and latter organizations can be grouped as manufacturing industry set.

Table 6.6. Client Characteristics for Period-End Closing Use Case. While *client2* and *client4* are operating in service industry, latter clients are active in manufacturing industry. Active SAP/CO components alter due to this domain variety.

candi	*Domain*	*Location*	*Number of Plants*	*Active SAP Components*	*Number of Traces*	*Number of Activities*	*Number of Transitions*	*Number of Connectors*
client1	*Cement Industry*	*Baku, AZ*	6	*Cost Center Accounting (CCA)* *Product Costing (PC)* *Material Ledger (ML)* *Plant Maintenance (PM)*	400	34	48	10
client2	*Higher Education*	*Baku, AZ*	2	*Cost Center Accounting (CCA)* *Internal Order Accounting (IO)* *Material Ledger (ML)*	400	15	20	4
client3	*Airspring Production*	*Bursa, TR*	4	*Cost Center Accounting (CCA)* *Product Costing (PC)* *Material Ledger (ML)* *Plant Maintenance (PM)*	400	27	41	8
client4	*Retail Marketing*	*Istanbul, TR*	10	*Cost Center Accounting (CCA)* *Material Ledger (ML)*	400	11	13	2
client5	*Automotive Industry*	*Bursa, TR*	3	*Cost Center Accounting (CCA)* *Product Costing (PC)* *Material Ledger (ML)*	400	23	36	8

The activity vocabulary consists of 45 activities that are valid at the reference business processes and the activityID is the concatenation of three codes: (i) original SAP transaction code, (ii) sub-step of the underlying transaction code (e.g. CRE-Costing Run Creation, SEL-Selection, DTR-Sequence Determination, SNG-Single Level Price Determination, MLT-Multi Level Price Determination, MRK-Mark Material Price, REV-Post Closing Reverse, RVL-Revaluation of Consumption/Settlement, INT-Initial settlement) and (iii) execution variant for the underlying transaction code (e.g. MNT-Maintenance, PRD-Production, CRS-Courses). Table 6.7 summarizes the valid activities in the form of SAP transaction code, sub-step and execution variant.

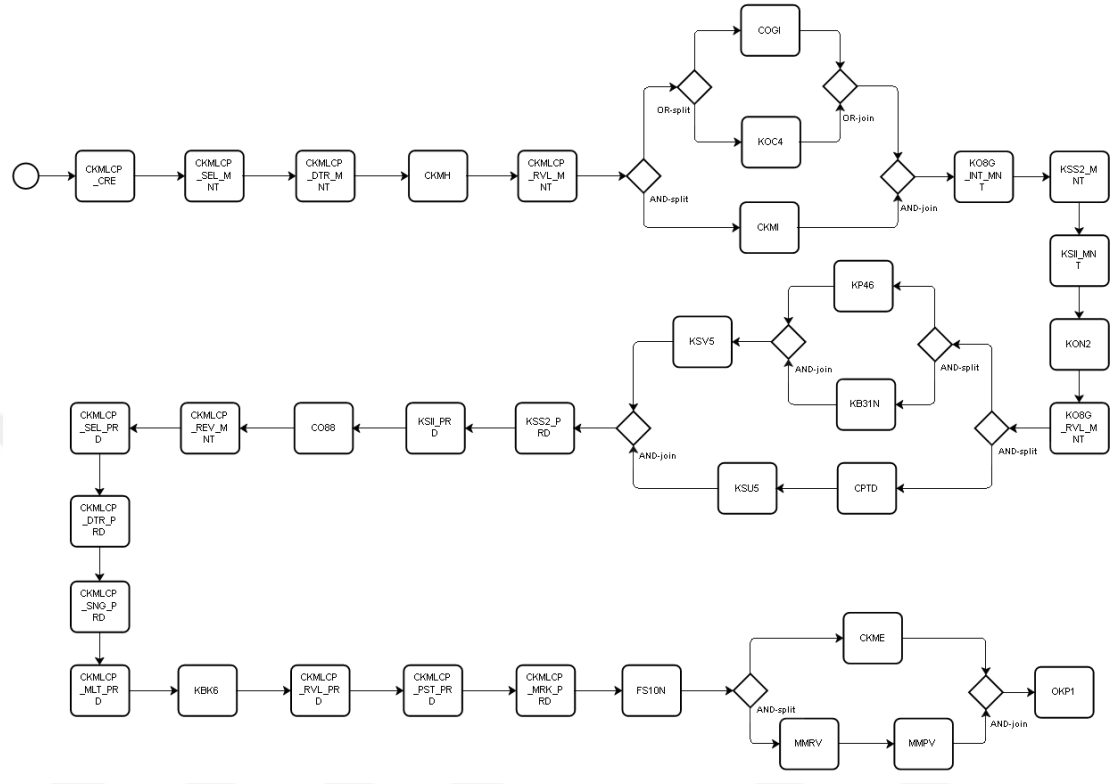
Table 6.7. Activity Dictionary for Period-End Closing Use Case. *activityID* is the concatenation of SAP transaction code, sub-step and execution variant occurred at reference process maps. Each transaction code is assigned to at least one SAP/CO component.

ActivityID	Description	SAP Transaction	Substep	Variant	SAP/CO Component
CKMDUVMAT	Distribution of Physical Inventory Differences	CKMDUVMAT			ML
CKME	Activation of Planned Prices	CKME			ML
CKMH	Single-Level Price Determination	CKMH			ML
CKMI	Post Closing	CKMI			ML
CKMLCP_CRE	Cockpit Actual Costing	CKMLCP	CRE		ML
CKMLCP_DTR_MNT	Cockpit Actual Costing	CKMLCP	DTR	MNT	ML
CKMLCP_DTR_PRD	Cockpit Actual Costing	CKMLCP	DTR	PRD	ML

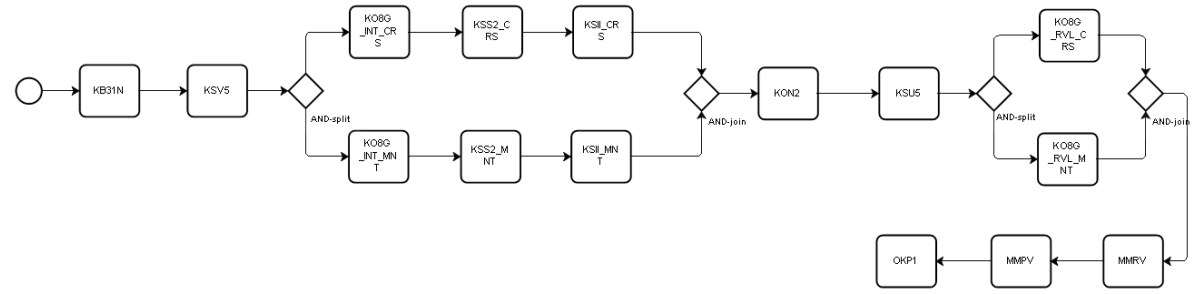
CKMLCP_MLT_PRD	Cockpit Actual Costing	CKMLCP	MLT	PRD	ML
CKMLCP_MRK_PRD	Cockpit Actual Costing	CKMLCP	MRK	PRD	ML
CKMLCP_PST_PRD	Cockpit Actual Costing	CKMLCP	PST	PRD	ML
CKMLCP_REV_MNT	Cockpit Actual Costing	CKMLCP	REV	MNT	ML
CKMLCP_RVL_MNT	Cockpit Actual Costing	CKMLCP	RVL	MNT	ML
CKMLCP_RVL_PRD	Cockpit Actual Costing	CKMLCP	RVL	PRD	ML
CKMLCP_SEL_MNT	Cockpit Actual Costing	CKMLCP	SEL	MNT	ML
CKMLCP_SEL_PRD	Cockpit Actual Costing	CKMLCP	SEL	PRD	ML
CKMLCP_SNG_PRD	Cockpit Actual Costing	CKMLCP	SNG	PRD	ML
CO88	Actual Settlement: Production/Process Order	CO88			PC
COGI	Processing Goods Movements with Errors	COGI			PC
CPTD	Actual Template Allocation: Production Order	CPTD			PC
FS10N	Balance Display	FS10N			CCA
KB31N	Enter Statistical Key Figures	KB31N			CCA
KBK6	Manual Actual Price	KBK6			CCA
KKAO	WIP Calculation: Collective Processing	KKAO			PC
KO8G_INT_CRS	Actual Settlement: Internal/Maintenance Orders	KO8G	INT	CRS	IO
KO8G_INT_MNT	Actual Settlement: Internal/Maintenance Orders	KO8G	INT	MNT	PM
KO8G_RVL_CRS	Actual Settlement: Internal/Maintenance Orders	KO8G	RVL	CRS	IO
KO8G_RVL_MNT	Actual Settlement: Internal/Maintenance Orders	KO8G	RVL	MNT	PM
KOC4	Cost Analysis: Internal/Maintenance Orders	KOC4			PM/IO
KON2	Actual Revaluation: Internal/Maintenance Orders	KON2			PM/IO
KP46	Change Statistical Key Figure Plan Data	KP46			CCA
KSII_CRS	Actual Price Determination: Cost Centers	KSII		CRS	IO
KSII_MNT	Actual Price Determination: Cost Centers	KSII		MNT	PM
KSII_PRD	Actual Price Determination: Cost Centers	KSII		PRD	PC
KSS2_CRS	Actual Cost Splitting: Cost Centers	KSS2		CRS	IO
KSS2_MNT	Actual Cost Splitting: Cost Centers	KSS2		MNT	PM
KSS2_PRD	Actual Cost Splitting: Cost Centers	KSS2		PRD	PC
KSU5	Execute Actual Assessment	KSU5			CCA
KSV5	Execute Actual Distribution	KSV5			CCA
ME23N	Display Purchase Order	ME23N			CCA
MMPV	Close Periods	MMPV			CCA
MMRV	Allow Posting to Previous Period	MMRV			CCA
OKP1	Maintain Period Lock	OKP1			CCA

As the starting point, reference process models, which are designed as *to-be process models* at business blueprints, are analyzed in order to get the insight about proposed business processes. Even though the reference processes are slightly different due to the active SAP/CO components and their

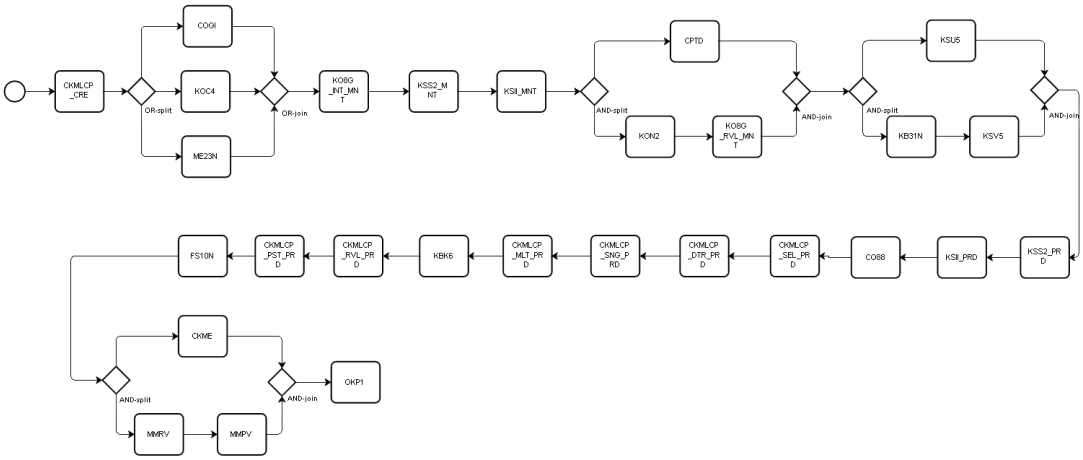
configurations, each reference process is initiated by CKMLCP_CRE (Cockpit Actual Costing–Costing Run Creation) or KB31N (Enter Statistical Key Figures) activities and terminated by OKP1 (Maintain Period Lock). Figure 6.5 depicts reference process maps for underlying clients.



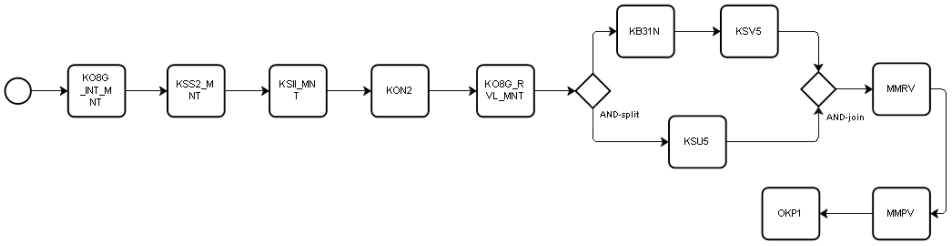
Client1 Process Model



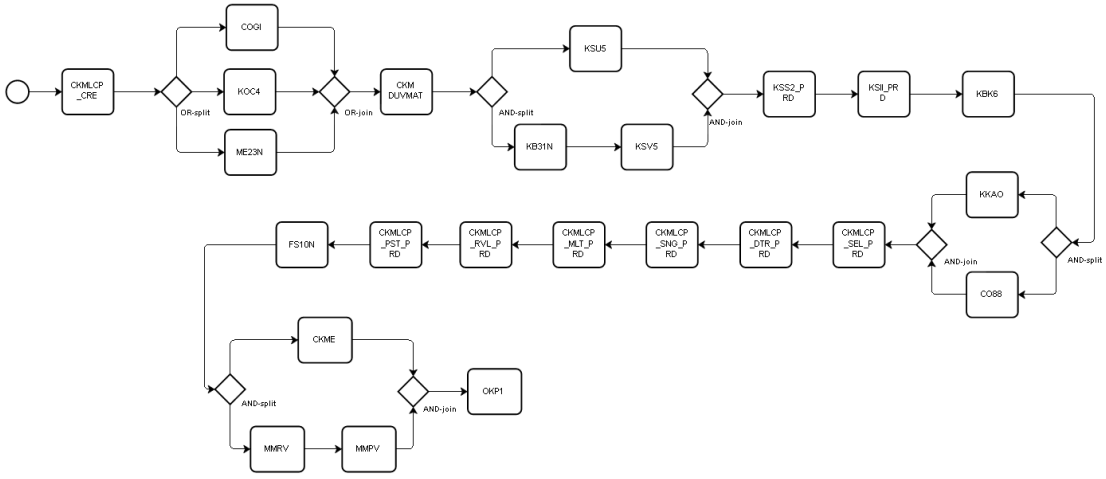
Client2 Process Model



Client3 Process Model



Client4 Process Model



Client5 Process Model

Figure 6.5. Proposed Process Maps for Process Alternatives (*clienti*) for Period-End Closing Use Case.

6.2.2. Synthetic Event Log Generation

As stated in [47], when we apply process mining techniques to ERP systems, we confront the problem about collecting the operational data, i.e. event logs, from the source system: The transactional logs of ERP system are not appropriate to monitor the individual cases (or process instances). Instead, ERP system only monitors the execution logs of specific transactions without any relation within a case identifier [47]. Additionally, ERP systems are highly *data-centric*, i.e. the transactional data is staggered through the operational database tables. Although SAP tools like Reverse Business Engineer (RBE) log the transaction frequencies and these transactions are linked to event process chain (EPC) formatted reference process models, they cannot be assigned to individual cases [47]. Due to these limitations, we develop a program, that automatically generates synthetic event logs for the given process model structure and process profile according to *Petri net firing rule* as follows [10]:

***Definition* (Petri net)**. A Petri net is a triple (*P, T, F*) such that,

- *P* is a finite set of *places.* A place *p* is called an *input place* of a transition *t* if there exists a directed arc from *p* to *t*. Otherwise it is called an *output place* of transition *t* if there exists a directed arc from *t* to *p*.
- *T* is a finite set of *transitions*.
- *F* is a set of arcs (*flow* relation).

At any time a place contains any tokens, it is shown as black dot [10]. The *state*, called as marking, holds the current distribution of these tokens over places and the number of existing tokens is varying. The marking procedure of Petri net is defined as *Petri net firing rule* as follows [10]:

- A transition *t* is said to be *enabled* if each input place *p* contains at least one token.
- An enabled transition may fire. If a transition *t* fires, then t consumes one token from each input place *p* and *produces* one token for each output place *p* of *t*.

In addition to the *Petri net firing rule*, automated event log generation requires two data lists: (i) activity list and (ii) Petri net list. The activity list holds the *activity type* (i.e. I-initiator, O-ordinary, C-connector and S-sink) and the *priority* (i.e. the probability of activity occurrence changing between 0 and 100) per activity. Petri net list converts the graph-based process model into tabular format, in which each transition is enlisted as *predecessor*, *successor* and *transition type* (i.e. AND/OR/XOR join or split-type gateway and direct succession). Table 6.8 exemplifies the activity list and Petri net for the reference process model given in Figure 6.2.

Table 6.8. Activity List and Petri Net Lists for Reference Process Model. The activity list holds the activity type and priority (e.g. there happens an XOR branching after ADVN activity is fired. CONF activity will be tokenized with 90% priority). Petri net list converts the graph-based process model into a tabular form.

Activity list

Activity ID	Actvitiy Type	Priority
CREA	I	100
ADVN	O	100
CONF	O	90
CNCL	S	10
CMPL	O	100
EXPS	O	100
DISP	O	100
CHCK	O	100
ACCN	S	100

Petri net

Predecessor	Successor	Transition Type
CREA	ADVN	*direct succ*
ADVN	CONF	*XOR-split*
ADVN	CNCL	*XOR-split*
CONF	CMPL	*direct succ*
CMPL	EXPS	*direct succ*
EXPS	DISP	*AND-split*
EXPS	CHCK	*AND-split*
DISP	ACCN	*AND-join*
CHCK	ACCN	*AND-join*

According to Petri-net's firing rule and the data list requirements, synthetic event logs are generated as follows:

i. As the starting point, the initiator activity (I-typed) of the underlying business process (e.g. CREA-Travel request create) is fired and the successors of the initiator (ADVN-Advance payment) are tokenized.
ii. Then one of the tokenized activities is selected according to the priority value at activity list and then randomly selected activity is fired. Due to this firing, the successors of fired activity are tokenized.
iii. If fired activity is connected to its predecessor by an AND-split, then other AND-splitted successors are highly prioritized at the next firing step. For XOR-split option, the unfired successors are suppressed for the current process cycle. On the other hand, OR-splitted successors are conditionally fired according to *OR threshold*.
iv. The AND-joined activity waits for all predecessors to be fired and tokenized. Then it propagates the tokenization to subsequent successor(s). This tokenization and firing iteration is continued up to a sink-typed (S-typed) activity is fired.

In addition to this algorithm, various parameters are used for handling specific conditions:

- *Surprise effect* is used in order to call unexpected process cycle terminations. For instance, bankruptcy is a niche case in banking financial processes and this relatively least probable case can be taken into consideration by the surprise effect.
- *Noise factor* is used to generate noisy event logs that deteriorate the Petri-net firing rule.
- OR-split gateway specifies that one or more of the tokenized successors will be fired in the case of OR-split. *OR threshold* reflects this conditional firing.

6.3. Dominant Behavior Extraction Analysis

This phase mainly aims to find out the dominant behavior, which refers to the common subsequence of activities recurring across process instances with certain domain significance. This domain information is a kind of generalization for the process knowledge that is highlighted at the event logs. According to tacit process variant assumption, which states the fact there may be more than one process variant in a single event log, and there is no available knowledge on how to partition the set of cases [48], there may arise an inductive biasness at dominant behavior extraction. To minimize this bias, 25 consecutive runs are performed with varying process discovery and Genetic Algorithms parameter settings and 25 versions of dominant behaviors per process alternative are extracted. The details about corresponding process discovery and Genetic Algorithms parameter settings are given in Section 5.1.4.

6.3.1. Process Discovery Based Analysis

There are various efforts in the literature towards measuring the quality of process models. In [52], Vanderfeesten et al. aim to adapt software engineering quality metrics (e.g. coupling, cohesion, complexity, modularity and size) into business process modeling domain as guiding principles. Additionally according to [53], it is important to realize the fact that there is never a single learned model for a given event log, since there are syntactically different process models having similar behaviors and an infinite number of models can be discovered for a given set of event logs [17, 20]. In addition to this rationale, process discovery algorithms have to handle the following issues:

- *Dealing with incompleteness.* Incompleteness is the anomaly that reflects only a part of the process behavior is observed at the event logs. Since the number of interleavings among concurrent activities increases in an exponential fashion, total completeness is an impractical assumption [53].
- *Further abstraction*. For relatively complex business processes, discovering a spaghetti-like process model is more probable. Instead of possible exceptions, i.e. *process veins* in [20], the main process flow, i.e. *process arteries* in [20], might be focused to deal with *noise* that is a rare or infrequent behavior not representative for typical behavior [53]. In this aspect of abstraction, although interpretability is increased, this leads to models with low *precision* and *fitness* values.

In the context of these four main quality dimensions (generalization, simplicity, precision and fitness), we design a conformance checker that supports two metrics for judging the quality of process discovery at dominant behavior extraction phase: *completeness* and *soundness*.

Definition **(Completeness)**. Completeness of the process model *PM* is the fraction of the traces in the event log that may be the result of some enactment at the corresponding process model [30] as given in Equation 6.1.

$$completeness(PM, L_P) = \frac{\left|\{s \in L_P \wedge s \in PM\}\right|}{\left|\{s \in L_P\}\right|} \tag{6.1}$$

In this aspect, the completeness metric is similar to *fitness* in [23, 89] and *recall* in [19, 50].

Definition **(Soundness)**. A totally complete model may support not only the traces provided in the event logs, but also an arbitrary number of execution patterns that are registered. Such a condition can be measured by another metric named *soundness*. Soundness measures possible process enactments at the corresponding process model *PM* that find some correspondence in the event logs [30].

$$soundness(PM, L_P) = 1 - \frac{\left|\{s \in L_P \wedge s \in PM\}\right|}{\left|\{s \in PM\}\right|} \quad (6.2)$$

The soundness metric is similar to *minimality* or *behavioral appropriateness* in [24] and *precision* in [19, 23, 50]. Theoretically, high soundness implies that many activities in the process model have limited correspondence in the event logs. At completeness calculation, these corresponding *free activities* cause a biased high completeness than they are attained to an event in the event log [11].

As shown in in Figure 6.6, having high completeness and low soundness values, extracted dominant behaviors capture most of the events in the process logs at Travel Management use case. However, dominant behavior can be partially captured for process alternative *cand3*. This is possibly due to high variation with the process executions due to high connectivity and low density characteristics of process alternative *cand3* and high sensitivity to confidence and support thresholds.

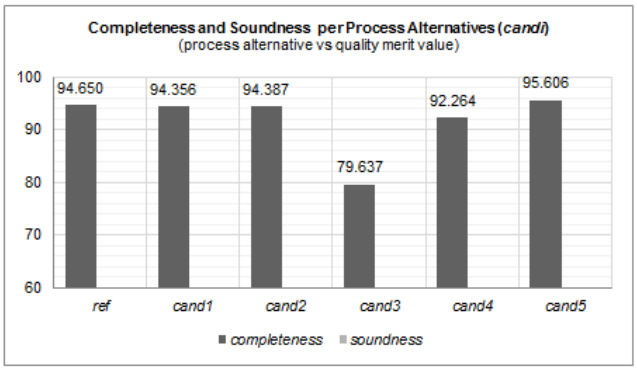


Figure 6.6. Average Completeness and Soundness Values per Process Alternatives (*candi*) for Travel Management Use Case.

For Period-End Closing use case, process alternatives with an average 97.3% completeness and 0-valued soundness levels highlight the fact that; the underlying process discovery mechanism shows a good balance between completeness and soundness quality metrics according to Figure 6.7.

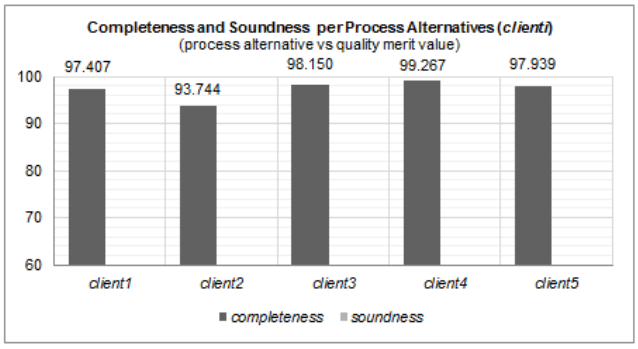


Figure 6.7. Average Completeness and Soundness Values per Process Alternatives (*clienti*) for Period-End Closing Use Case.

In addition to completeness and soundness metrics, we aim to analyze the understandability of the process model in structural perspective. This *structural influence factor set* is composed of *connectivity*, *density* and *average transition length* (ATL):

- *Connectivity* is the average number of transitions (edges) per activity (node) at discovered or proposed process model (i.e. |T|/|A|).
- *Density* is the ratio between total number of activities (nodes) and total number of blocks (activities and AND/OR/XOR gateways) at discovered or proposed process model (i.e. |A|/(|A|+|C|)).

- *Average transition length* (ATL) is the average length per transition at discovered or proposed process model. The unit length between two adjacent activities is assumed as 1 unit for this factor computation.

While connectivity and density factors resemble *coupling* and *cohesion* design principle introduced in [52], average transition length measures the compactness of the dominant behavior, lower average transition length converging to 1 value implies a compact dominant behavior, i.e. dominant behavior that tends towards assigning activities with relatively stronger interactions at neighboring positions. Table 6.9 summaries the structural factors per process alternative.

Table 6.9. Structural Factors per Process Alternatives for Travel Management Use Case.

candi	*Number of Activities*	*Number of Transitions*	*Number of Connectors*	*Connectivity*	*Density*	*Average Transition Length*
ref	9	9	3	**1.00**	**0.75**	**4.55**
cand1	9	9	3	**1.00**	**0.75**	**4.55**
cand2	9	9	3	**1.00**	**0.75**	**4.55**
cand3	9	10	3	**1.11**	**0.75**	**4.77**
cand4	9	9	4	**1.00**	**0.69**	**4.74**
cand5	9	9	3	**1.00**	**0.75**	**4.55**

Potentially, process alternative *cand3* is more vulnerable to changes at confidence and support threshold according to higher connectivity, since process alternatives with higher connectivity tends to have a higher likelihood towards *spaghetti-like models* and this feature increases the risk of pruning down by confidence or support threshold. Likewise, process alternative *cand4* with lower density is also sensitivity to these thresholds. Hence the change at [0.3, 0.5] confidence interval results in the loss of process behavior at process discovery and significant reductions at completeness as shown in Figure 6.8. Stabilization after 0.5 confidence threshold value means that only core process behaviors (i.e. direct successive typed transitions introduced at section 4.1.1) are left for the process alternatives except *cand5.* As a result, this mechanism results in lower average completeness as shown in Figure 6.6.

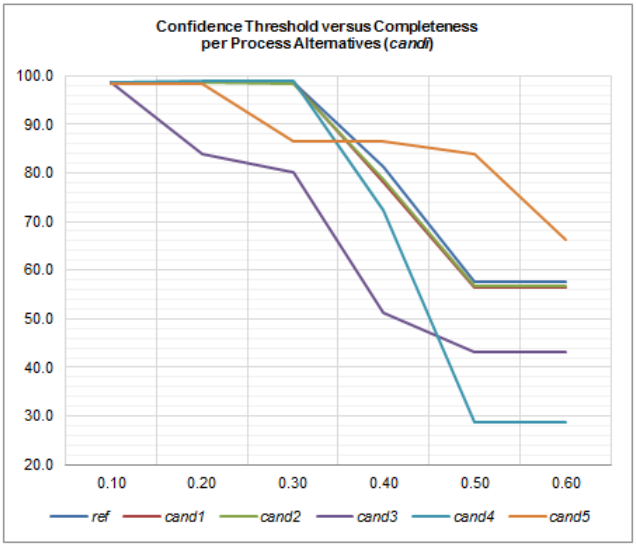


Figure 6.8. Confidence Threshold versus Completeness per Process Alternatives (*candi*) for Travel Management Use Case (*X-axis:confidence threshold, Y-axis:completeness*). Respectively, more *spaghetti-like* process alternatives (e.g. *cand3* and *cand4*) are more vulnerable for the changes at confidence threshold. *Higher connectivity* or *lower density* refers to *weak* transitions and these process behaviors are pruned down easily by the increase of confidence threshold value.

Respectively at Period-End Closing use case, *lasagna-like* process alternatives, which are characterized by higher density and lower connectivity metrics, are more robust to the increase of confidence threshold. As shown in Table 6.10 and Figure 6.9, process alternative *client4* is a compact

candidate with 1.18 connectivity and 0.85 density values. These characteristics indicate that there exists hardly any AND/OR/XOR connectors (gateways) and activities are mostly connected with direct-successive transitions. Additionally, 1.14 average transition length (ATL) also emphasizes this rationale such that, relatively minimal ATL values refer to a process model connected with only direct-successive transitions. On the other hand, process alternatives *client3* and *client5* tend to be *spaghetti-like* processes with lower density and higher connectivity. Weak-order transitions at these process alternatives are vulnerable to be pruned down by the increase of confidence and support threshold. Hence this pruning down diminishes the coverage of the dominant behavior and completeness.

Table 6.10. Structural Factors per Process Alternatives for Period-End Closing Use Case.

clienti	*Number of Traces*	*Number of Activities*	*Number of Transitions*	*Number of Connectors*	*Connectivity*	*Density*	*Average Transition Length*
client1	400	34	48	10	**1.41**	**0.77**	**1.48**
client2	400	15	20	4	**1.33**	**0.79**	**1.47**
client3	400	27	41	8	**1.52**	**0.77**	**1.54**
client4	400	11	13	2	**1.18**	**0.85**	**1.14**
client5	400	23	36	8	**1.57**	**0.74**	**1.50**

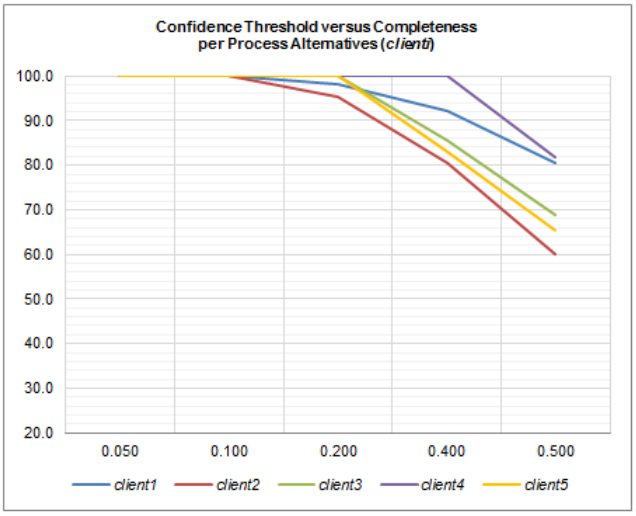


Figure 6.9. Confidence Threshold versus Completeness per Process Alternatives (*clienti*) for Period-End Closing Use Case. Respectively more *lasagna-like* process alternatives (e.g. *cand4*) are more robust to the changes at confidence threshold. On the other hand, the effect of the changes at confidence threshold is significant for *spaghetti-lie* process alternatives (i.e. *client3* and *client5*) according to the decreasing trend at completeness curves.

Process discovery analyses for other use cases are given in Appendix B.

6.3.2. Genetic Algorithms Based Analysis

Another enhancement at dominant behavior extraction phase is the *GA engine adaptation* that aims to find the dominant behavior with the *fittest solution*. Unlike to prior brute-force approach introduced in [42], three drivers are analyzed to interpret the performance and robustness of GA engine in Section 6.3.2: *schema application*, *crossover probability* and *population size*.

One of the most popular researches in Genetics Algorithms field has been done by Holland based on *Schema Theory* [22]. Accordingly, it is assumed that *good schemata* characteristic has a important effect on the individual's high fitness score and the likelihood of obtaining better approximations to the

underlying problem increases by inheriting the characteristics of these good schemata at the following populations [78, 79, 80]. If we focus on the effects of crossover and mutation framework on schema application, the likelihood of individuals with higher fitness will increase exponentially due to the effect of schema application and vice versa [78].

In this aspect, we attempt to handle the motivation of schema in Genetic Algorithms application within the context of Travel Management use case. Hence the process discovery runs for process alternative *cand1* are analyzed with or without schema runtime configuration[9]. According to Figures 6.10 and 6.11, the individuals encoded with a predefined schema receive an exponentially increasing number of trials while the number of individuals with less fit schemata will decrease in successive generations tremendously. Hence a *relatively rapid start phase* is detected for the process discovery runs with schema (i.e. average fitness score series evolves to the maximum series more rapidly for the runs with schema) and population convergence requires less iteration due to the schema.

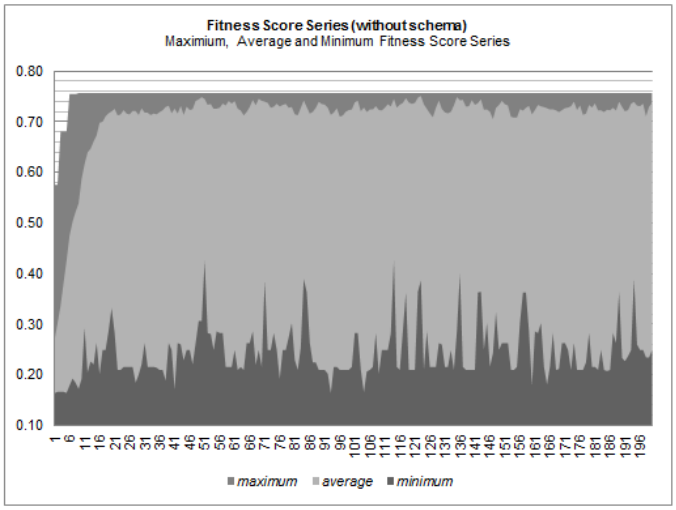


Figure 6.10. Maximum, Average and Minimum Fitness Score Series (without schema) for Travel Management Use Case.

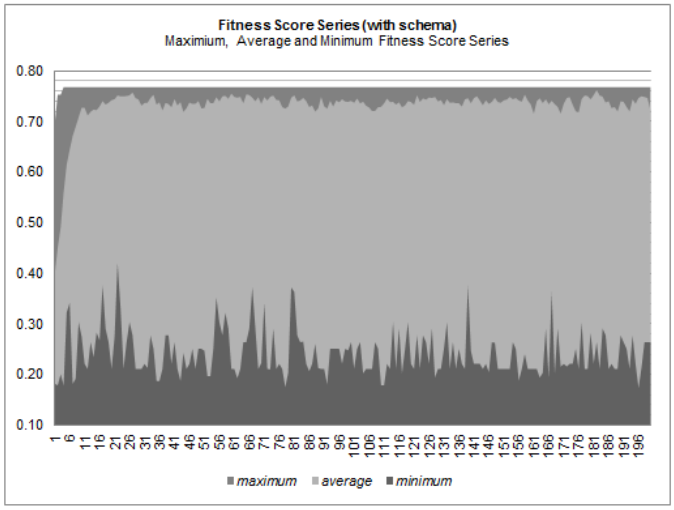


Figure 6.11. Maximum, Average and Minimum Fitness Score Series (with schema) for Travel Management Use Case. Since schema is a sub-pattern of gene values which inherits more characteristics of higher fitness score, a relatively rapid ramp-up phase is detected for the process discovery runs with schema application. Hence the convergence of average fitness series to maximum requires less iteration due to the schema.

[9]In addition to schema application, process discovery and GA parameters are configured as follows:
confidence/support threshold=0.15, backtracking penalty point=2, population size=150, P(crossover)=0.80

In addition to visualization of process discovery runs, it is aimed to statistically analyze the effect of schema by comparing difference between maximum and average fitness scores ($max_{fitness}$–$avg_{fitness}$) by *dependent t-test*. According to the t-value (-3.684 versus $t_{0.05,59}$), the null hypothesis, H_0, which states that there is no clear distinction between average fitness scores series for the process discovery run with or without schema, is rejected. Negative outcome implies that; initialization with a schema has a positive affect towards generating higher fitness score at next populations. The result of t-test (α=0.05 and CI=95%) is given in Table 6.11.

Table 6.11. Dependent t-test for Schema Application at Dominant Behavior Extraction with respect to the $max_{fitness}$–$avg_{fitness}$ Values for Travel Management Use Case.

Dependent t-Test Results for Schema Application

	with Schema	*without Schema*
Mean	0.03125	0.04674
Variance	0.00112	0.00433
Observations	60	60
Pearson Correlation	0.94700	
Hypothesized Mean Diff.	0.00000	
DF	59	
t Stat	**-3.68432**	
P(T<=t) one-tail	0.00000	
t Critical one-tail	1.67112	
P(T<=t) two-tail	**0.00136**	
t Critical two-tail	**2.00112**	

The main criticism of schema theory is the assumption that ignores the effect of crossover and mutation on the genetic variation. While this assumption is not a reasonable generalization, schema theory does not lead to any valid inference about the variations of population fitness over evolution iterations [22]. For this reason, we aim to analyze the effect of crossover at population convergence. While selection and schema application are respectively *conservative operators* that intend to reduce the diversity of population and simplify the content of population, crossover and mutation framework tends to increase the diversity of the corresponding population [79].

As shown in Figure 6.12, the process discovery runs at Travel Management use case with relatively low probability of crossover may be more conservative by traversing only a sub-region of the search space. This rationale can be realized by the slow start phase of maximum or average fitness series and the fluctuations at the minimum fitness series. On the other hand as shown in Figure 6.13, the runs with high probability of crossover frequently take larger steps in exploring the search space early by the effect of diversity at the initial population. Then smaller improvements occur when most individuals are quite similar at the corresponding population. This rationale can be seen at the steady-state phase of the average fitness series and lessen fluctuations at the minimum fitness series.

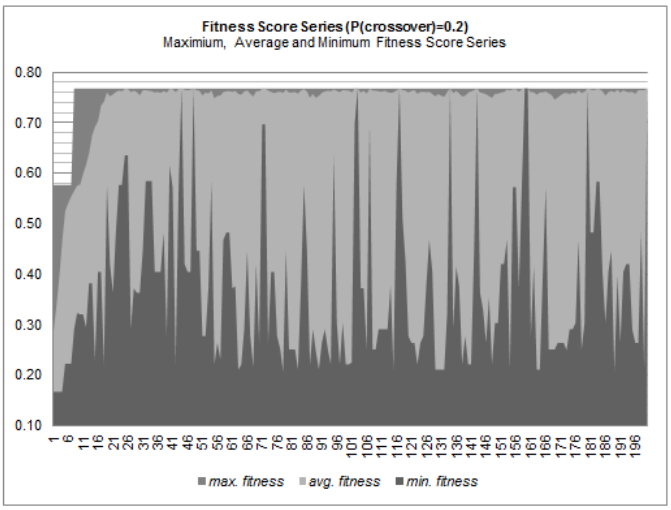


Figure 6.12. Maximum, Average and Minimum Fitness Score Series (P(crossover)=0.2) for Travel Management Use Case.

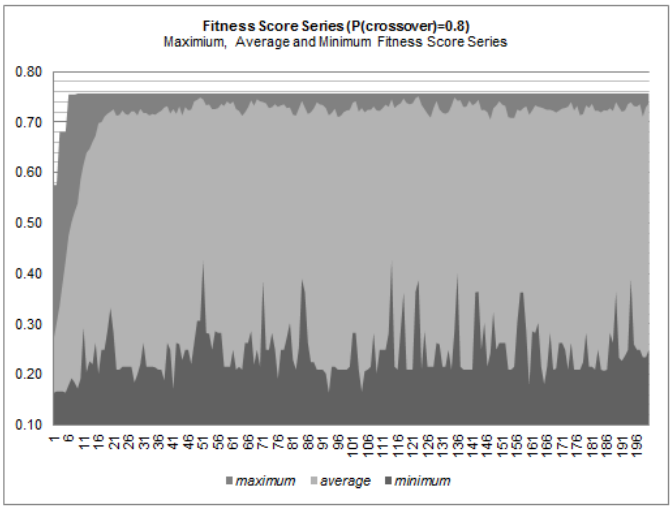


Figure 6.13. Maximum, Average and Minimum Fitness Score Series (P(crossover)=0.8) for Travel Management Use Case. Underlying process discovery problem might have several peak points (local optima) in search space. Unlike to other *myopic local search algorithms* (e.g. hill-climbing search), GA tends to abandon inefficient local optimality by the undirected jumps triggered due to the crossover operation.

As a result, process discovery runs with lower crossover probability and without schema application are tend to behave like myopic local search, since there is a risk to stagger at local optima regions and premature population convergence.

In parallel to this outcome, we also aim to analyze the effect of population size at population convergence. Generally speaking, the larger the training dataset the better process modeling, although the returns begin to diminish once a certain volume of training data is exceeded [89]. Similarly population size encourages the offspring selection to utilize the available genetic information in the current population to the maximum level in terms of achieving new and even better solution candidates for the successive generations [57]. For typical Genetic Algorithms applications, the suggested population size is between 10-160 individuals [30]. Hence population size affects the efficiency and performance of Genetic Algorithms.

In this aspect, we attempt to handle the effect of population size in Genetic Algorithms application in the context of Loan Application use case. Hence process discovery runs for process alternative *cand1* are analyzed with distinct population size settings, i.e. pSize=100 versus pSize=500. According to Figure 6.14, lower population size configuration guides the process discovery runs to generate poor solutions. This rationale can be realized by the slow start phase of maximum or average fitness series and the fluctuations at the minimum fitness series. On the contrary as in Figure 6.15, runs with larger population size frequently take larger steps in exploring the search space early by the effect of genetic diversity at the initial population. Then smaller improvements occur when most individuals are quite similar at the corresponding population. But the use of larger populations does not always improve the solution accuracy and only increases required computational resources.

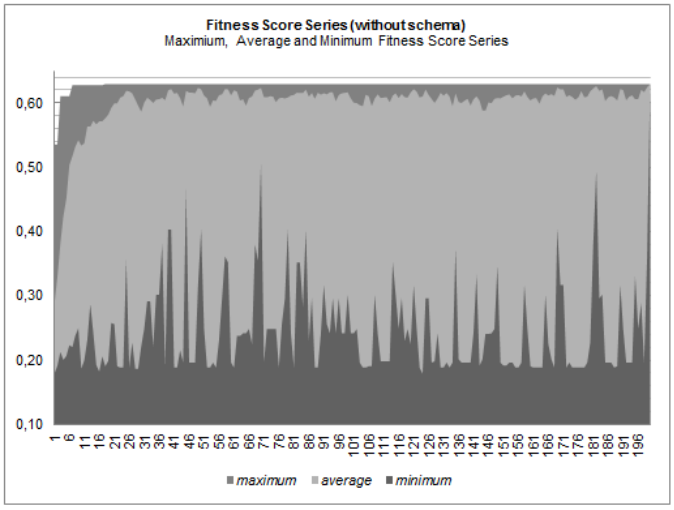


Figure 6.14. Maximum, Average and Minimum Fitness Score Series (pSize=100) for Loan Application Use Case.

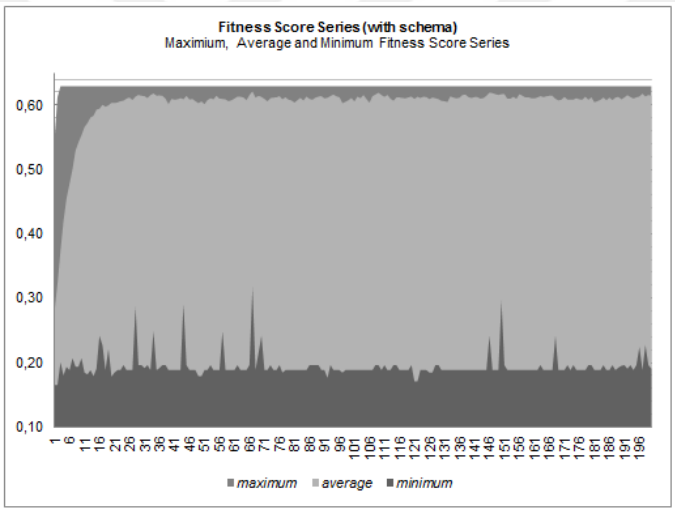


Figure 6.15. Maximum, Average and Minimum Fitness Score Series (pSize=500) for Loan Application Use Case.

In parallel to visualization of process discovery runs, it is aimed to statistically analyze the effect of population size by comparing average fitness scores ($avg_{fitness}$) by *dependent t-test*. According to the t-value (-2.212 versus $t_{0.05,199}$), the null hypothesis, H_0, which states that there is no clear distinction between average fitness scores series for the process discovery run due to population size, is rejected. Negative outcome implies that; process discovery runs with larger population size tends to generate stronger individuals with higher fitness scores for the next generations. The result of t-test (α=0.05 and CI=95%) is given in Table 6.12.

Table 6.12. Dependent t-test for Population Size at Dominant Behavior Extraction with respect to the $avg_{fitness}$ Values for Loan Application Use Case.

Dependent t-Test Results for Population Size

	pSize=100	*pSize=500*
Mean	0.60012	0.04674
Variance	0.00112	0.00174
Observations	200	200
Pearson Correlation	0.96981	
Hypothesized Mean Diff.	0.00000	
DF	199	
t Stat	**-2.21183**	
P(T<=t) one-tail	0.01413	
t Critical one-tail	1.65254	
P(T<=t) two-tail	**0.02815**	
t Critical two-tail	**1.97225**	

Runtime information (i.e. dominant behavior sequence, completeness, soundness, average transition length, average transition number per activity and total process time) about dominant behavior extraction phase for all use cases is given in Appendix C.

According to the results of process discovery based analysis, process candidates with *higher connectivity* and *lower density* tend to generate *spaghetti-like process models* that are hard to interpret by process observers. Therefore this characteristic increases the risk of pruning by confidence/support threshold and results in the loss of process behavior at process discovery. Unlike to spaghetti-like process models, *lasagna-like* process candidates with *lower connectivity* and *higher density* are more robust to the increases at confidence threshold. Moreover, *lower ATL* highlights the mechanism with respect to compactness, which is encouraged by assigning the activity pairs with stronger succession to consecutive neighboring positions at the sequence.

Moreover, the rationale hindered by Holland's schema theorem is validated by statistical tests such that, process discovery runs with schema requires less iterations to reach to the population convergence according to the difference between the maximal and average fitness scores. Alternatively, process discovery with lower crossover probability and limited population size tend to behave like myopic local search due to the risk of congesting at local optimal points. On the contrary, opposite GA configuration has a better performance in exploring the search space by the effect of genetic diversity. Then smaller improvements happen when most individuals become quite similar at the following populations.

6.4. Sequence Alignment Analysis

6.4.1. Single-Reference Pairwise Alignment Based Analysis

As expressed in the *cetaris paribus* rule, which means with other conditions remaining the same; other things being equal, Single-Reference Pairwise Alignment is designed by fixing one of the process alternatives as *reference* and altering the latter process alternatives as *candidate* iteratively. Accordingly, the confidence tables and extracted dominant behaviors are inherited from dominant behavior extraction phase and corresponding similarity scores are calculated in the context of two components as introduced in Section 5.2.2:

- The value change at similarity score due to replacement (match or mismatch) move is handled as *structural similarity* (*strSim*). This implicitly conserved region is the evidence of common functionality and business context overlapping between process alternatives.
- The value change at similarity score due to inDel (insertion and deletion) move is handled as *behavioral similarity* (*bhvrSim*). The regions that are rarely filled with gap symbol (−) emphasize the deviations and exceptional behaviors among the process alternatives.

Total, structural and behavioral similarity scores for Travel Management use case are given in Figures 6.16, 6.17 and 6.18 respectively. Visually, process alternative *cand3* is the most similar candidate to the *reference*. While structural similarities of process alternatives *cand2, cand3* and *cand5* are too closed, the major distinction is in the result of behavioral similarities, which are totally negative. This negative behavioral similarity scores highlights the rationale such that, insertion or deletion operation is strongly discouraged or penalized by the business context, which is encoded by the confidence tables.

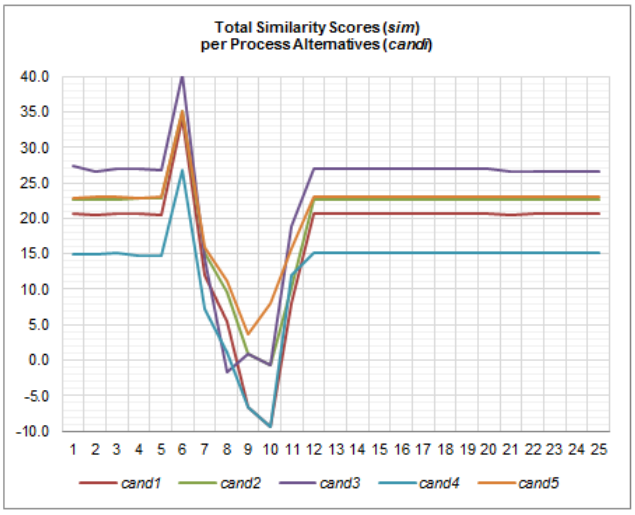


Figure 6.16. Total Similarity Score (*sim*) per Single-Reference Pairwise Alignment Run and Process Alternatives (*candi*) for Travel Management Use Case (*X-axis:alignment runID, Y-axis:similarity score*).

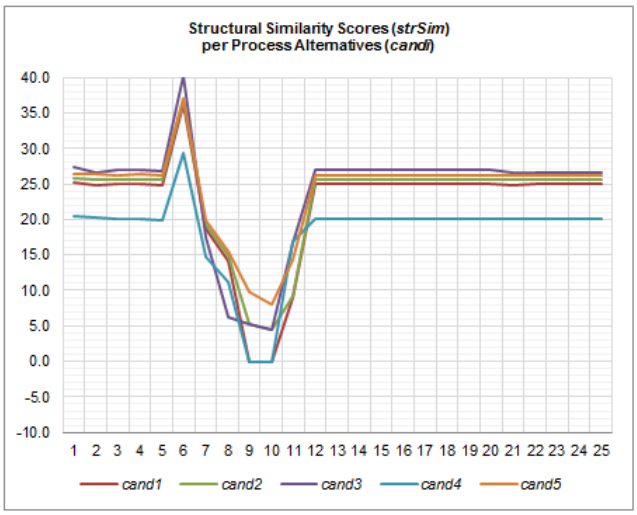


Figure 6.17. Structural Similarity Score (*strSim*) per Single-Reference Pairwise Alignment Run and Process Alternatives (*candi*) for Travel Management Use Case (*X-axis:alignment runID, Y-axis:structural similarity score*).

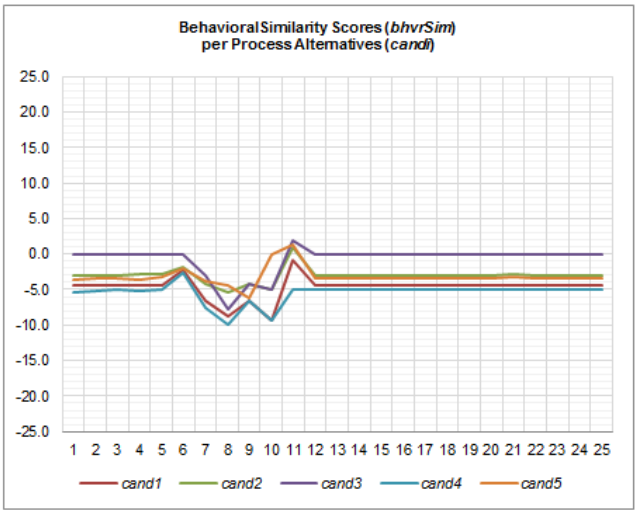


Figure 6.18. Behavioral Similarity Score (*bhvrSim*) per Single-Reference Pairwise Alignment Run and Process Alternatives (*candi*) for Travel Management Use Case (*X-axis:alignment runID, Y-axis:behavioral similarity score*).

In addition to Single-Reference Pairwise Alignment technique, process similarity measurement is performed by prior alignment approaches, i.e. NW-classical Needleman Wunsch and CANW-Confidence-aware Needleman Wunsch algorithms, to validate the leverage effect of the confidence enhanced cost functioning introduced in Section 5.2.2. In [66], Esgin and Karagoz initially aim to demonstrate that process mining can benefit from the sequence mining techniques, which are strengthened with standard Needleman-Wunsch algorithm (NW) to quantify the similarities and discrepancies. This prior algorithm evaluates the dominant behaviors by avoiding the requirement for well-structured process models. Edit distance metric values in [66] are set to confidence from-to chart threshold (*confThr*) value. Afterwards, a new alignment approach called CANW-Confidence-aware Needleman Wunsch is introduced in [67]. According to CANW, inDel scores are determined with respect to the interactions among activities. Hence, these insertion and deletion operations are dynamically determined in a context-aware fashion. While, these interactions are figured out by confidence metric (confFTC) given at Equation 4.2 and case-based inDel scores are calculated for each iteration by this metric. On the other hand, default match (or mismatch) values are set to the confidence from-to chart threshold (*confThr*) value. Figures 6.19 and 6.20 show the total similarity scores for Travel Management use case that are measured by CANW and NW approaches respectively.

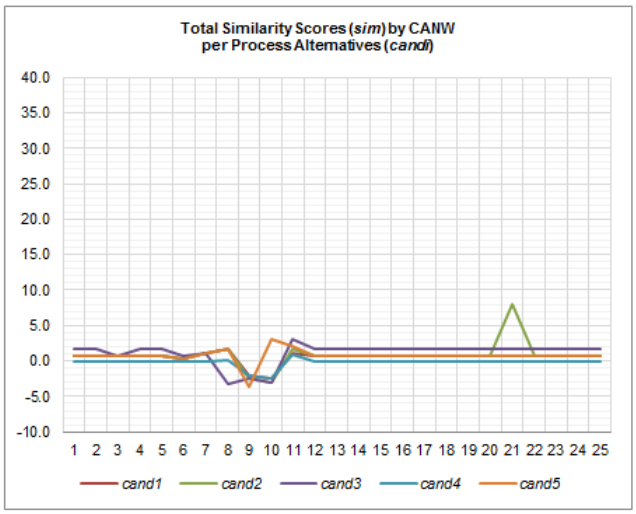


Figure 6.19. Total Similarity Score (*sim*) per CANW Run and Process Alternatives (*candi*) for Travel Management Use Case (*X-axis:alignment runID, Y-axis:similarity score*).

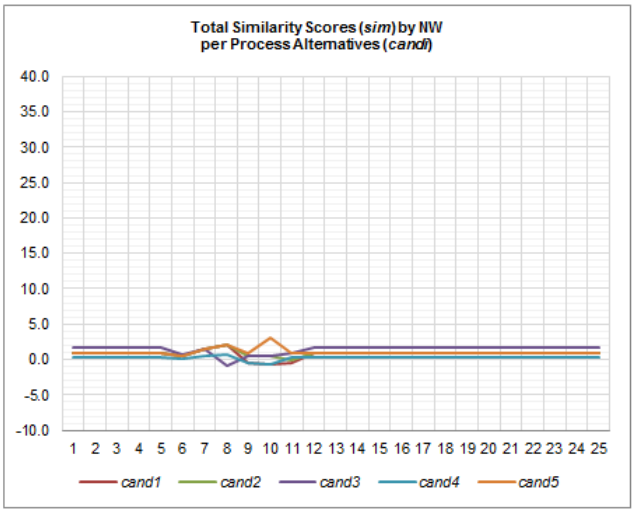


Figure 6.20. Total Similarity Score (*sim*) per NW Run and Process Alternatives (*candi*) for Travel Management Use Case (*X-axis:alignment runID, Y-axis:similarity score*).

In parallel to Travel Management use case, total, structural and behavioral similarity scores for Loan Application use case are measured by Single-Reference Pairwise Alignment runs as shown in Figures 6.21, 6.22 and 6.23 respectively. Visually, process alternative *cand4* is the most similar candidate to *reference*. While structural similarities of process alternatives *cand2* and *cand3* dominate the similarity measurements, the major distinction is the result of behavioral similarities, which are totally negative. This negative behavioral similarity scores highlight the mechanism such that, *lasagna-like* process alternatives with limited activity vocabulary are more conservative to the insertion or deletion operations that violate the business context. On the other hand, *high connectivity* and *low density* feature of process alternative *cand4* make it possible to evolve the process structure towards the reference, since this characteristic lessens the sparsity of the confidence table and tends to generate positive inDel scores.

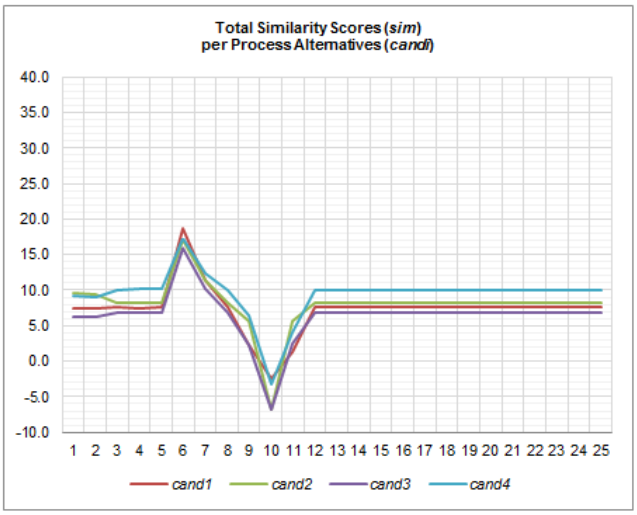


Figure 6.21. Total Similarity Score (*sim*) per Single-Reference Pairwise Alignment Run and Process Alternatives (*candi*) for Loan Application Use Case (*X-axis:alignment runID, Y-axis:similarity score*).

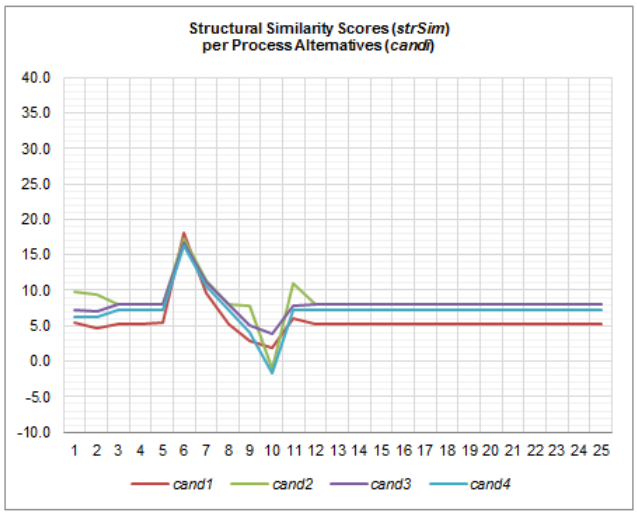


Figure 6.22. Structural Similarity Score (*strSim*) per Single-Reference Pairwise Alignment Run and Process Alternatives (*candi*) for Loan Application Use Case (*X-axis:alignment runID, Y-axis:structural similarity score*).

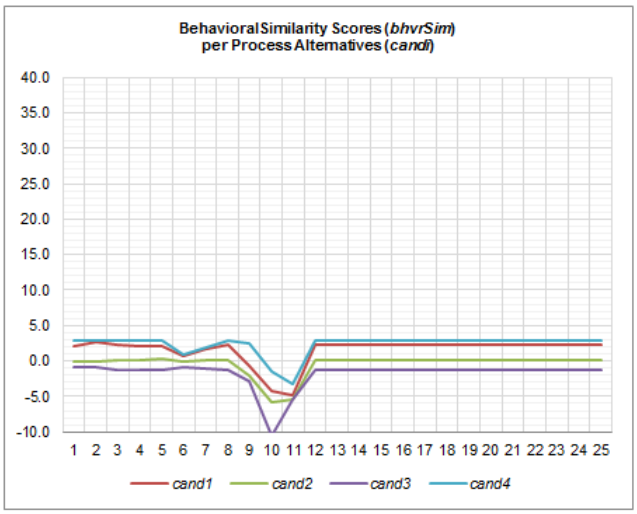


Figure 6.23. Behavioral Similarity Score (*bhvrSim*) per per Single-Reference Pairwise Alignment Run and Process Alternatives (*candi*) for Loan Application Use Case (*X-axis:alignment runID, Y-axis:behavioral similarity score*).

Figures 6.24 and 6.25 show the total similarity scores for Loan Application use case that are measured by CANW and NW approaches. As seen in the figures, the value range of similarity scores for these prior approaches shrinks to [-4.0, 4.0] interval due to the limitations and incapabilities within prior cost functioning such that, confidence metric (confFTC) may provide limited deviations and diversity at valuating the edit operations.

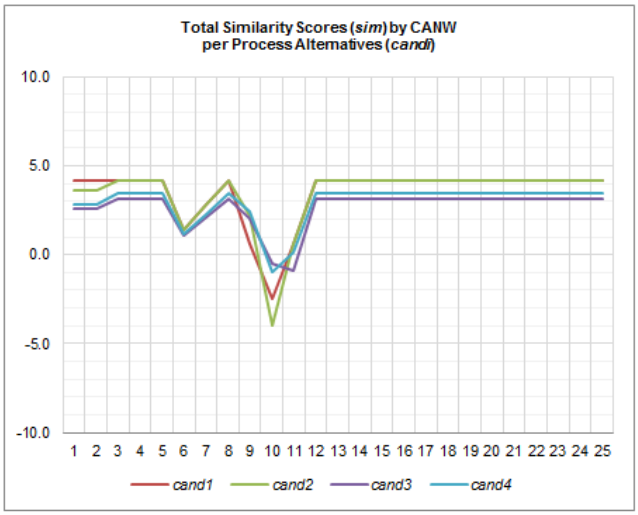


Figure 6.24. Total Similarity Score (*sim*) per CANW Run and Process Alternatives (*candi*) for Loan Application Use Case (*X-axis:alignment runID, Y-axis:similarity score*).

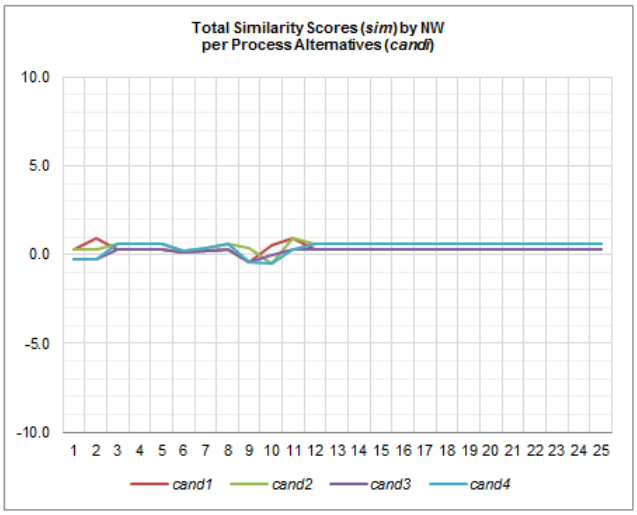


Figure 6.25. Total Similarity Score (*sim*) per CANW Run and Process Alternatives (*candi*) for Loan Application Use Case (*X-axis:alignment runID, Y-axis:similarity score*).

6.4.1.1. Intuitive Judgment Based Analysis

The validation of alignment approaches is based on the comparison of similarity measurements with the intuitive judgments of professional process observers specialized on various ERP systems. Accordingly, various information retrieval (IR) metrics and recall/precision framework are adapted into process similarity measurement. Then we elaborate on the accuracy of these metrics in comparison to intuitive judgment based validation. Indeed, handling of these judgments as the *ground truth* or benchmark in process similarity measurement is also preferred in previous studies such as [30, 91]. Basically, the intuitive judgments were collected by a questionnaire, at which the process maps of reference and candidate process alternatives are listed. 25 process observers (or domain experts with different process expertise) visually analyzed and ranked the process alternatives according to the similarity to the reference. Then these responses are converted to *likert-chart*, which are the tables summarizing the likelihood rankings of each candidate process alternatives (c_i) with respect to the reference. Additionally, the likelihood rankings of the process alternatives are also measured according to prior NW adaptations. Underlying rankings for Travel Management use case are transformed into the likert-charts as shown in Table 6.13.

Table 6.13. Likert Charts for Intuitive Judgments (IJ) and Alignment Approaches for Travel Management Use Case. Column indexes (e.g. *p=1 the most and p=5 the least similar process alternative*) denote trial numbers for relevant likelihood ranking position *p*. *Weight* coefficient holds the average likelihood of corresponding process alternative, *ci,* and *rank* is the nominal position for *ci*.

*IJ-Intuitive Judgments (**ground truth**)*

	1	2	3	4	5	Σ	weight	rank
c1	3		1	17	4	25	3.76	4
c2	3		19	3		25	2.88	3
c3	18	1	2	1	3	25	1.80	1
c4	1	3	2	4	15	25	4.16	5
c5	1	21			3	25	2.32	2

Pairwise Sequence Alignment

	1	2	3	4	5	Σ	weight	rank
c1			1	22	3	25	4.06	4
c2		6	18	1		25	2.80	3
c3	21		3		1	25	1.40	1
c4			1	3	22	25	4.82	5
c5	4	19	2			25	1.92	2

CANW-Confidence Aware Needleman Wunsch

	1	2	3	4	5	Σ	weight	rank
c1	3		21	1		25	2.80	3
c2	2		22		1	25	2.92	4
c3	22		1		2	25	1.40	1
c4	2		1		22	25	4.60	5
c5	3	1	20		1	25	2.80	2

NW-Needleman Wunsch

	1	2	3	4	5	Σ	weight	rank
c1	2		20		3	25	3.08	3
c2	2		22	1		25	2.88	2
c3	22		2		1	25	1.32	1
c4		1	15		9	25	3.68	4
c5	5		6		14	25	3.72	5

According to the rank values in Table 6.13, there emerges a perfect commonality between the intuitive judgments and Pairwise Alignment in the first glance. These results are then evaluated under information retrieval (IR) related metrics, i.e. cosine similarity and discounted cumulative gain (DCG). In order to apply cosine similarity, we priorly construct *likert-chart vectors* that convert the results at the likert-charts to vector format.

***Definition* (likert-chart vectors).** Let LC_x be a likert-chart of alignment approach x holding relative rankings of each process alternative c_i. For alignment approach x we define *likert-chart vector* lc_x such that, term weight lc_{xi} is the *weight coefficient* of process alternative c_i at likert-chart LC_x. Table 6.14 summarizes the likert-chart vectors per alignment approaches.

Table 6.14. Likert-Chart Vectors (*lcx*) per Alignment Approach and Term Weights (*lcxi*) for Travel Management Use Case.

		term weight (lc_{xi}) per process alternative (c_i)				
Likert Chart Vector (lc_x)	Alignment Approach	c1	c2	c3	c4	c5
lc1	IJ	**3.76**	**2.88**	**1.80**	**4.16**	**2.32**
lc2	Pairwise Sequence Alignment	4.06	2.80	1.40	4.82	1.92
lc3	CANW	2.80	2.92	1.40	4.60	2.80
lc4	NW	3.08	2.88	1.32	3.68	3.72

***Definition* (cosine similarity).** Let lc_x and lc_y be the corresponding likert-chart vectors for alignment approaches x and y respectively. *Cosine similarity* between approaches x and y denoted by $cosSim(x,y)$ is cosine of the angle between those likert-chart vectors, lc_x and lc_y as given in Equation 6.3. The value of $cosSim(x,y)$ ranges from –1 (quite distinct) to +1 (equivalent).

$$cosSim(x,y) = \frac{\vec{lc}_x \times \vec{lc}_y}{|\vec{lc}_x| \cdot |\vec{lc}_y|} = \frac{\sum_{j=1}^{N} \vec{lc}_{xj} \cdot \vec{lc}_{yj}}{\sqrt{\sum_{j=1}^{N} \vec{lc}_{xj}^{2}} \cdot \sqrt{\sum_{j=1}^{N} lc_{yj}^{2}}} \tag{6.3}$$

According to cosine similarity values given in Figure 6.26, Pairwise Alignment is the most similar approach to the intuitive judgments with a similarity value of 0.993. Likewise, process observers rank the process variants according to the similarity insights with the reference process alternative for Loan Application use case. Then these rankings and the likelihood rankings of prior NW adaptations are converted to the likert chart as given in Table 6.15.

Table 6.15. Likert Charts for Intuitive Judgments (IJ) and Alignment Approaches for Loan Application Use Case.

*IJ-Intuitive Judgments (**ground truth**)*

	1	2	3	4	Σ	weight	rank
c1	11	1	12	1	25	2,12	2
c2		17	2	6	25	2,56	3
c3		2	5	18	25	3,64	4
c4	14	5	6		25	1,68	1

Pairwise Sequence Alignment

	1	2	3	4	Σ	weight	rank
c1	4	1	19	1	25	2,68	3
c2	3	20	1	1	25	2,00	2
c3		1	3	21	25	3,80	4
c4	18	3	2	2	25	1,52	1

CANW-Confidence Aware Needleman Wunsch

	1	2	3	4	Σ	weight	rank
c1	25				25	1,00	1
c2		2	19	4	25	3,08	3
c3		1		24	25	3,92	4
c4	1	22		2	25	2,12	2

NW-Needleman Wunsch

	1	2	3	4	Σ	weight	rank
c1	4			21	25	3,52	3
c2	23	1		1	25	1,16	1
c3		1		24	25	3,92	4
c4	20			5	25	1,60	2

These results are then evaluated under cosine similarity according the likert-chart vector transformation. As shown in Table 6.16, we priorly construct the likert-chart vectors and then apply cosine similarity given in Equation 6.3.

Table 6.16. Likert-Chart Vectors (*lcx*) per Alignment Approach and Term Weights (*lcxi*) for Loan Application Use Case.

		term weight (lc_{xi}) per process alternative (c_i)			
Likert Chart Vector (lc_x)	Alignment Approach	c1	c2	c3	c4
lc1	IJ	**2,12**	**2,56**	**3,64**	**1,68**
lc2	Pairwise Sequence Alignment	2,68	2,00	3,80	1,52
lc3	CANW	1,00	3,08	3,92	2,12
lc4	NW	3,52	1,16	3,92	1,60

According to cosine similarity values given in Figure 6.27, Pairwise Alignment is the most similar approach to the intuitive judgments with a 0.98 similarity value.

As an alternative metric, *discounted cumulative gain* (*DCG*) is a popular measure for evaluating the information retrieval and related tasks for evaluating web search [100]. It is based on two assumptions:

- Respectively, higher relevant documents are more practical than the marginally relevant document [100]. This relevance is figured by *graded relevance* given in Equation 6.4.
- According to logarithmic relation given in Equation 6.5, the usefulness of the relevant document is directly proportional to the ranked position [100], i.e. *higher i* value implies lower rank. This is due to the low probability to be examined.

Firstly, *graded relevance* (*rel*) as a metric of usefulness or gain is determined for each process alternative (c_i) by the *weight* coefficient values at intuitive judgments likert-chart (LC_{IJ}). In other words, intuitive judgments are evaluated as the *ground truth*.

$$rel_{ci} = 1 / weight_{ci,IJ} \qquad (6.4)$$

$$DCG_x = \sum_{i=1}^{p} \frac{2^{rel_{ci}} - 1}{\log_2(i+1)} \qquad (6.5)$$

As given in Equation 6.4, *graded relevance* (rel_{ci}) is the reciprocal of *weight coefficient* of process alternative *ci* at intuitive judgments (IJ) likert-chart. Graded relevance values and their distribution according to likelihood rankings for Travel Management use case are given in Tables 6.17.

Table 6.17. Graded Relevance (rel_i) Values per Intuitive Judgments and Alignment Approach for Travel Management Use Case. Alternative orders are determined according to the *rank* attribute in Table 6.13.

	ground truth (IJ)		Pairwise Sequence Alignment		CANW		NW	
i	alternative order	rel(i)	alternative order	rel(i)	alternative order	rel(i)	alternative order	rel(i)
1	**c3**	0.556	**c3**	0.556	**c3**	0.556	**c3**	0.556
2	**c5**	0.431	**c5**	0.431	**c5**	0.431	**c2**	0.347
3	**c2**	0.347	**c2**	0.347	**c1**	0.266	**c1**	0.266
4	**c1**	0.266	**c1**	0.266	**c2**	0.347	**c4**	0.240
5	**c4**	0.240	**c4**	0.240	**c4**	0.240	**c5**	0.431

***Definition* (discount cumulative gain).** DCG does highly relevant process alternatives appearing lower at the likelihood ranking tends to be penalized, since the graded relevance value diminishes logarithmically proportional to the position *i* at the ranking [100]. The DCG accumulated for a particular alignment approach *x* is given in Equation 6.5.

According to *DCG* values given in Figure 6.26, Pairwise Alignment is the most similar alignment approach to the intuitive judgments (*IJ*) with a 1.442 *DCG* value (and 1.0 *normalized DCG* value) for

Travel Management use case. Hence, it can be concluded that, Pairwise Alignment approach appropriately reflects the perceptions of process observers and there is a significant consistency between discount cumulative gain and cosine similarity metrics.

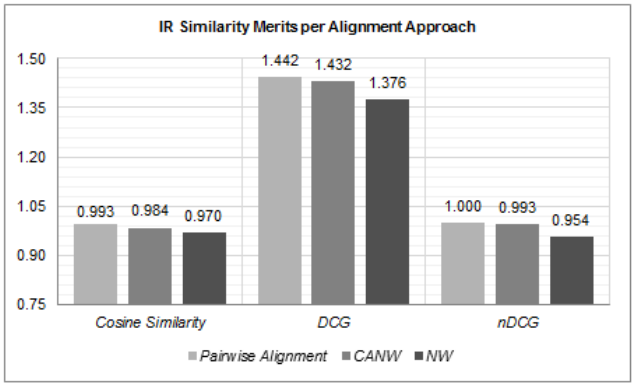


Figure 6.26. Similarity Metrics (c*osine similarity*, *DCG* and *normalized DCG*) per Alignment Approach for Travel Management Use Case. Pairwise Alignment challenges prior approaches according to relatively higher similarity values.

Alternatively, graded relevance values and their distribution according to likelihood rankings for Loan Application use case are given in Tables 6.18.

Table 6.18. Graded Relevance (rel_i) Values per Intuitive Judgments and Alignment Approach for Loan Application Use Case. Alternative orders are determined according to the *rank* attribute in Table 6.15.

	ground truth (IJ)		*Pairwise Sequence Alignment*		*CANW*		*NW*	
i	*alternative order*	*rel(i)*	*alternative order*	*rel(i)*	*alternative order*	*rel(i)*	*alternative order*	*rel(i)*
1	***c4***	0,595	***c4***	0,595	***c1***	0,472	***c2***	0,391
2	***c1***	0,472	***c2***	0,391	***c4***	0,595	***c4***	0,595
3	***c2***	0,391	***c1***	0,472	***c2***	0,391	***c1***	0,472
4	***c3***	0,275	***c3***	0,275	***c3***	0,275	***c3***	0,275

DCG given in Equation 6.5 has two advantages compared to other IR metrics. First, DCG allows each retrieved document has graded relevance while most traditional ranking measures only focus on binary relevance (i.e. relevant or irrelevant). Second, DCG implicates a discount function over the rank while other IR metrics uniformly weight all ranking positions. This diminishing value effect is reflected by the logarithmic denominator emphasized in Equation 6.5.

According to DCG values given in Figure 6.27, Pairwise Alignment approach is the most similar alignment approach to the intuitive judgments (IJ) with a 0.991 DCG value (and 0.99 normalized DCG value) for Loan Application use case. Hence, it can be concluded that; Pairwise Alignment appropriately reflects the perceptions of process observers.

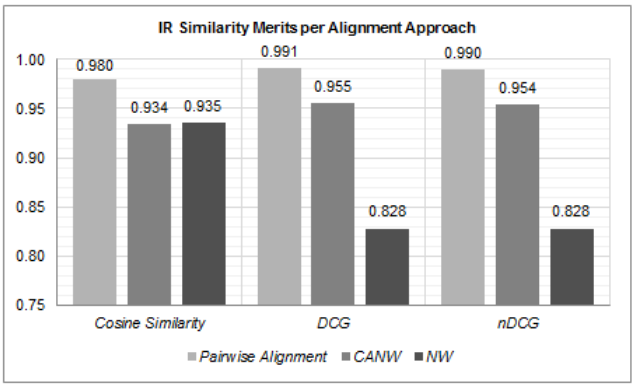


Figure 6.27. Similarity Metrics (c*osine similarity*, *DCG* and *normalized DCG*) per Alignment Approach for Loan Application Use Case. Pairwise Alignment challenges prior approaches according to relatively higher similarity values.

It is also possible to check the quality of the similarity rankings proposed by each alignment approach according to the responses of process observers. Similar to [91], two quality metrics are adapted from information retrieval domain: *recall* and *precision*.

***Definition* (recall).** *Recall* quantifies how much of the intuitive judgments (i.e. process alternative likelihood rankings, LR_{IJ}) is complied with the rankings of the underlying alignment approach *x* (LR_x). Recall metric is similar to *fitness* in [89].

***Definition* (precision).** *Precision* measures the ratio of the likelihood rankings belonging to the alignment approach *x* (LR_x) that finds some correspondence in the intuitive judgments (LR_{IJ}).

While *recall* value of alignment approach *x* for intuitive judgment *i*, i.e. $LR_{x,i}$, is the average value obtained with respect to *m* alignment runs, *precision* value of alignment approach *x* at alignment run *j*, i.e. $LR_{x,j}$, is the average value obtained with respect to *n* intuitive judgments. These metrics are given in Equations 6.6 and 6.7 respectively:

$$recall_{x,i} = \sum_{j=1}^{m} \left(LR_{x,j} \cap LR_{IJ,i} \right) \Big/ m \tag{6.6}$$

$$precision_{x,j} = \sum_{i=1}^{n} \left(LR_{x,j} \cap LR_{IJ,i} \right) \Big/ n \tag{6.7}$$

As shown in Figure 6.28, Pairwise Alignment that is closer to the top-right corner show a good balance between recall and precision for Travel Management use case in such a way that, it tends to repeat likelihood rankings proposed by process observers, while disregarding uncommon rankings.

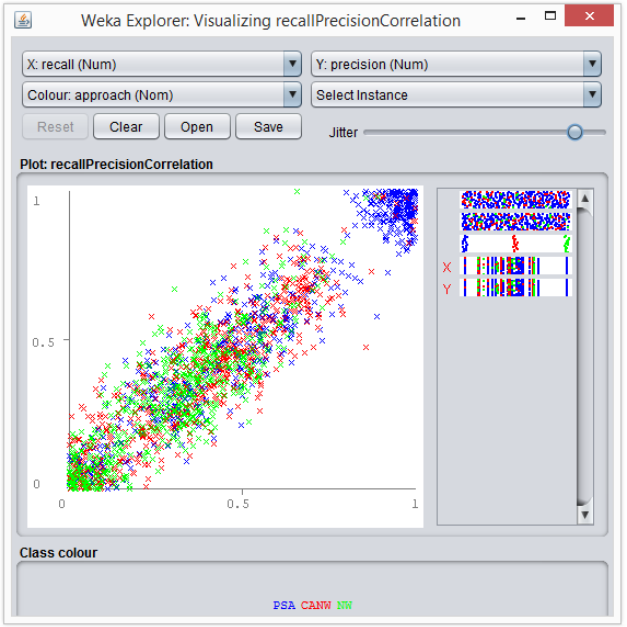


Figure 6.28. Recall versus Precision Correlation per Alignment Approach for Travel Management Use Case. Pairwise Alignment, i.e. the outermost blue series, performs better in terms of 0.602 average recall and precision values. On the other hand, the performances of the prior approaches, i.e. CANW and NW, are approximately 0.379 and 0.291.

Alternatively, it is aimed to analyze the correlation between the professional experience of process observers and the similarity measurement concern of the approaches. Hence as the *experience* factor, the participants were asked about their professional experience duration at business process modeling field and this influence factor is categorized at 4 levels: lead consultant (*LC*), senior consultant (*SC*), consultant (*C*) and assistant consultant (*AC*). When we solely focus on the responses belonging to lead (*LC*) and senior-level (*SC*) process observers by omitting the responses from less experienced participants, there occurs an improvement at precision values such that, *average precision* value of Pairwise Alignment is improved to 0.745 value (23.75% increase) and the discrepancy between Pairwise Alignment and prior alignment approaches is extended as shown in Figure 6.29. This outcome highlights that the similarity scoring at Pairwise Alignment significantly overlaps with the tacit similarity mechanism of expert level process observers.

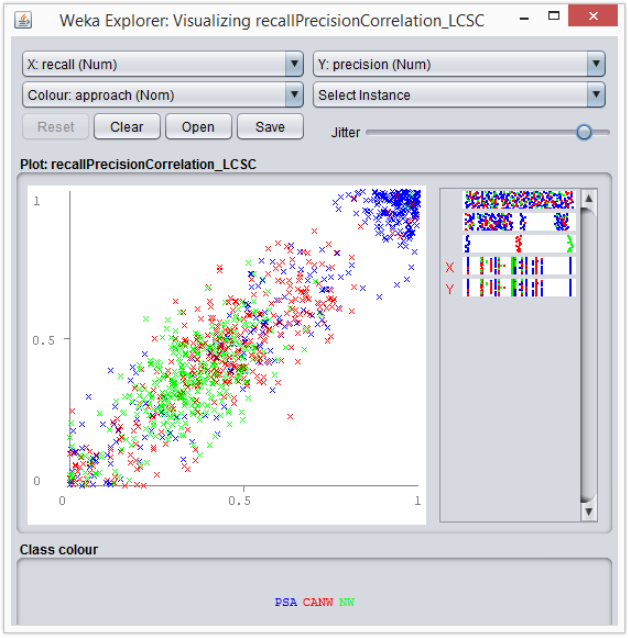


Figure 6.29. Plot of Precision Series per Alignment Approach (after *experience* factor analysis) for Travel Management Use Case. There is a significant consistency between the perceptions of experienced process observers and Pairwise Alignment approach such that, average recall (and precision) value of the corresponding approach is augmented from 0.602 to 0.745 value after eliminating the responses of *less-experienced* (AC and C-level) participant.

Similar to Travel Management use case, Pairwise Alignment that is closer to the top-right corner show a good balance of recall and precision for Loan Application use case as shown in Figure 6.30. It tends to repeat likelihood rankings proposed by the process observers, while disregarding uncommon rankings.

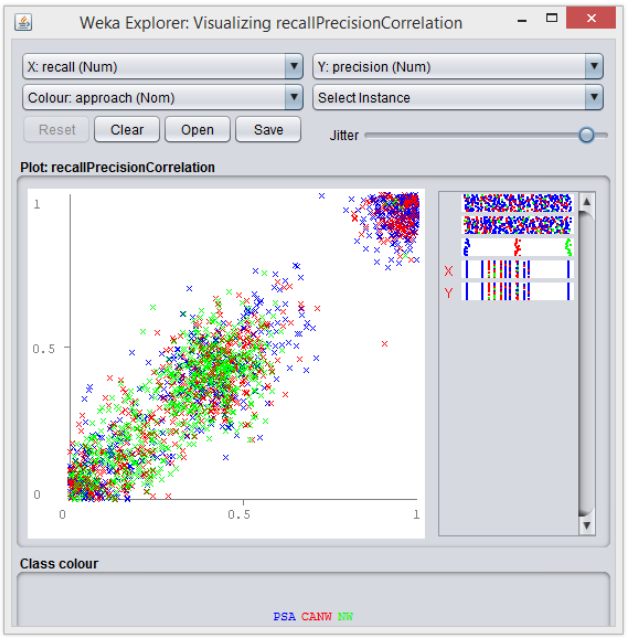


Figure 6.30. Recall versus Precision Correlation per Alignment Approach for Loan Application Use Case. Pairwise Alignment, i.e. the outermost series, performs better in terms of 0.534 average recall and precision values. On the other hand, the performances of the prior approaches, i.e. CANW and NW, are approximately 0.42 and 0.305.

In order to analyze whether there is a basic parallelism between the professional experience of process observers and similarity measurements, we majorly handle the process similarity ranking responses that are belonged to more experienced process observers (i.e. lead or senior-level consultants). When the recall/precision framework analysis dataset is reduced due to this omitting, *average precision* value of Pairwise Alignment is improved to 0.748 level and the discrepancy between Pairwise Alignment and prior approaches is extended as shown in Figure 6.31.

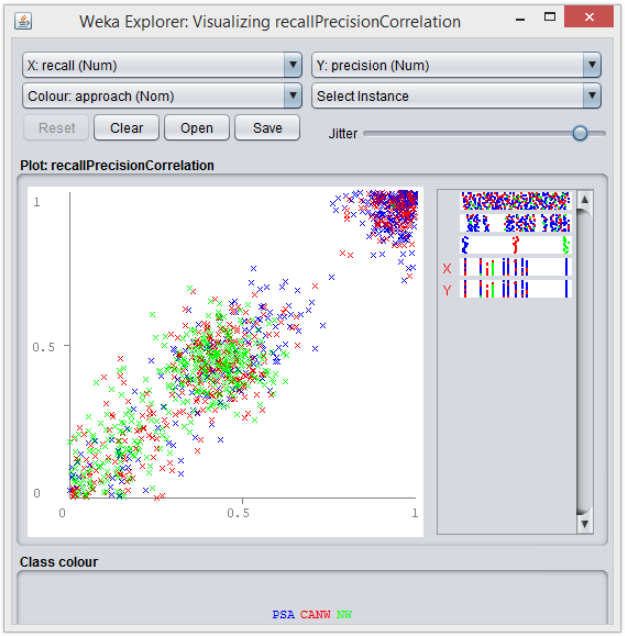


Figure 6.31. Plot of Precision Series per Alignment Approach (after *experience* factor analysis) for Loan Application Use Case. There is a significant consistency between the perceptions of experienced process observers and Pairwise Alignment approach such that, average precision value of the corresponding approach is augmented by approximately 40% after eliminating the responses of less-experienced (AC and C-level) participant.

Consequently, according to cosine similarity and discount cumulative gain metrics, Single-Reference Pairwise Alignment is highly correlated with the perceptions of process observers. According to recall and precision framework emphasized in [91], Single-Reference Pairwise Alignment shows a good balance according to this framework. Alternatively, there is a strong parallelism between the similarity scoring of Single-Reference Pairwise Alignment and the tacit similarity assessment mechanism of more experienced process owners due to significant improvement at the average precision value.

6.4.1.2. Comparison with Prior NW Adaptations

This section introduces the fundamental mechanism that determines the alignment of dominant behaviors and similarity scoring proposed in [66, 67]. Trying to estimate the effect of distance functions strictly from scores is sometimes insufficient and it might lead to wrong conclusions. To overcome these difficulties we propose a metric named *semantic similarity*, based on relative distances between distance functions concept in [57]. Semantic similarity compares the aligned dominant behavior sequences and it criticizes the identical pairs (*IPs*) repeats at these sequences by ignoring the similarity score values of these alignment operations.

***Definition* (semantic similarity).** Let $align_1$ and $align_2$ be two sequence alignments with respect to the two alignment approaches A_1 and A_2. Semantic similarity (*semSim*) is the ratio of *identical pairs* (i.e. *IP*, the same pair of activities positions at both alignments) to the total length of alignments, $align_1$ and $align_2$.

$$semsim(A_1, A_2) = \frac{2 \times |IP|}{|align_1| + |align_2|} \quad (6.8)$$

Respectively at Loan Application use case, the process alternatives with *higher connectivity* (connectivity≥1) and *lower density* (density≤1) features, e.g. *reference* and *cand4*, are structurally more spaghetti-like. These structural characteristics result in non-sparse confidence tables that are conjugated by AND/OR/XOR gateways and this status of confidence tables make the inDel operations more feasible and practical in sequence alignment respectively.

CANW is like the intermediate form at the progress from classical NW adaptation to Pairwise Alignment approach. This evolutionary progress emerges an inequality between match and inDel operations at confidence enhanced costing function such that, while inDel operations are highly sensitive to business context in terms of confidence values, match operation is fixed to the confidence for from-to chart threshold (*confThr*). Consequently, CANW is biased towards matching operation and Pairwise Alignment behaves like CANW due to similar inDel costing functions given in Equations 5.8, 5.9 and 5.10. As a result, *behavioral similarity scores* dominate the similarity measurements for spaghetti-like process alternative, i.e. *cand4*, as shown in Figure 6.23. Figure 6.32 highlights the semantic similarity between alignment approaches in the context of process alternative *cand4*.

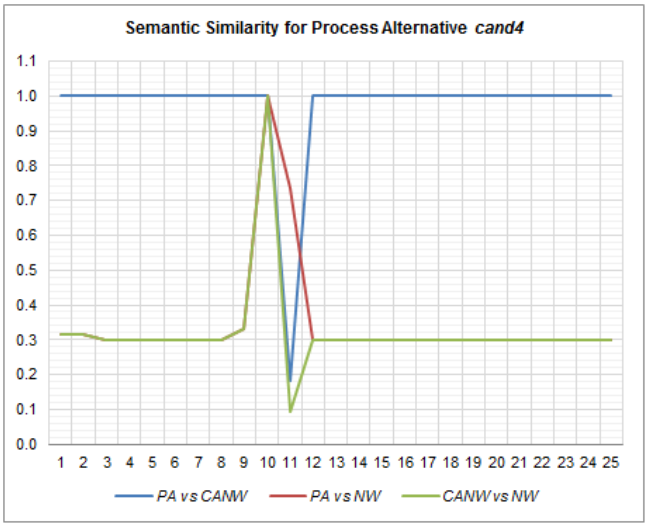


Figure 6.32. Semantic Similarity (*semSim*) for *cand4* at Loan Application Use Case (*X-axis:alignment runID, Y-axis:semantic similarity score*). Semantic similarity between Pairwise Alignment and CANW with an average value of 0.967 emphasize a strong overlapping in terms of identical pairs (IP).

Although process alternative *cand4* is one of the most similar process alternatives according to the intuitive judgments (IJ) and there is a strong semantic similarity between Pairwise Alignment and CANW, this likelihood is not observed in the context of IR-related metrics as shown in Figure 6.27. The underlying reason of this contradiction is the nature of semantic similarity such that, this metric is solely based on the *task label similarity*, which measures semantic and syntactic similarity based on various string edit distance and morphological analysis. Therefore it ignores magnitude of the similarity scores and the nominal likelihood rankings of these alignments. This sounds sensible, as the distribution of similarity scores is very much dependent on the balance between the cost function of matching and inDel operations.

Details about Single-Reference Pairwise Alignment (i.e. aligned forms of dominant behavior sequences per PA, CANW and NW approaches and similarity scores) are given in Appendix D.

6.4.2. Multi-Reference Pairwise Alignment Based Analysis

Pairwise Alignment is an adaptation of Needleman-Wunsch algorithm, which exploits the similarity scores between process alternatives on pairwise basis. While confidence enhanced cost function given at Equations 5.4 - 5.10 are also valid for Pairwise Alignment, this technique is performed at *only base-level* (*level=1*). As stated in section 6.1, Pairwise Alignment is performed in two options: *single-reference* and *multi-reference*. In Travel Management and Loan Application use cases, one of the process alternatives is fixed as *reference* and latter alternatives are selected as *candidate*. On the other hand, Multi-Reference Pairwise Sequence Alignment refers to a combinatorial reference selection (i.e. $C\binom{n}{2}$) such that, each process alternative is set as reference once and set as candidate for (n -1) times.

Figures 6.33 - 6.37 show the similarity scores per process alternative as reference for Environment Permit Application use case. As seen in Figure 6.36, the representation gap between *wabo4* and latter process alternatives emphasize the rationale that, *wabo4* process alternative tends to behave like a *singleton* that is quite different from other candidates. Additionally, *wabo4* is represented as the lowest trend at all figures except Figure 6.36. Although such a similarity gap is also valid for *wabo1* in Figure 6.33, *wabo1* normalizes this situation by converging to other process alternatives, i.e. *wabo2 and wabo3*, at the following multi-reference alignment runs as shown in Figure 6.34 and 6.35. Additionally, there is a strong commonality between process alternatives *wabo2* and *wabo3*, which results in a precipitated process family instantiation at process clustering.

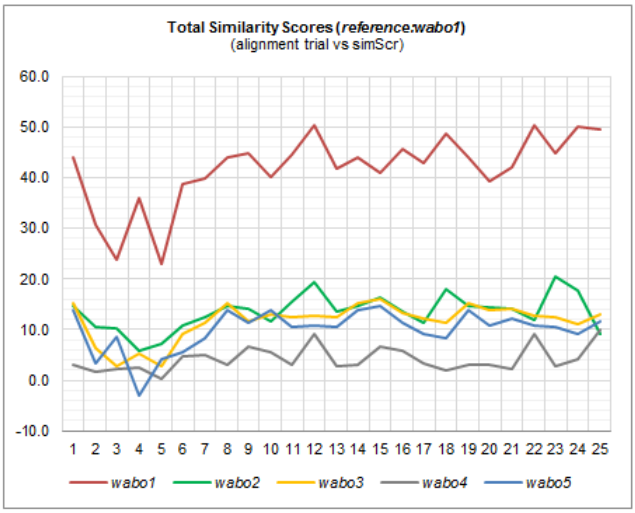


Figure 6.33. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo1*) (*X-axis:alignment runID, Y-axis:similarity score*).

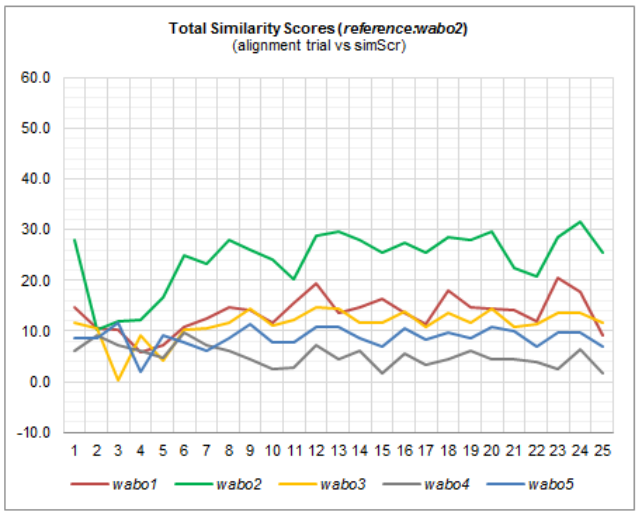


Figure 6.34. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo2*) (*X-axis:alignment runID, Y-axis:similarity score*).

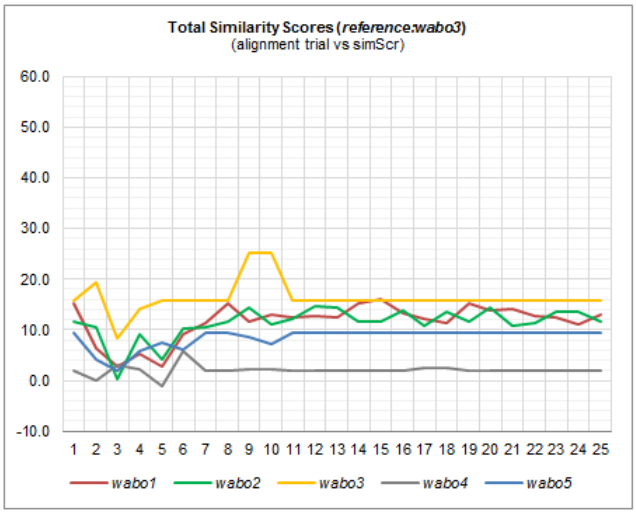


Figure 6.35. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo3*) (*X-axis:alignment runID, Y-axis:similarity score*).

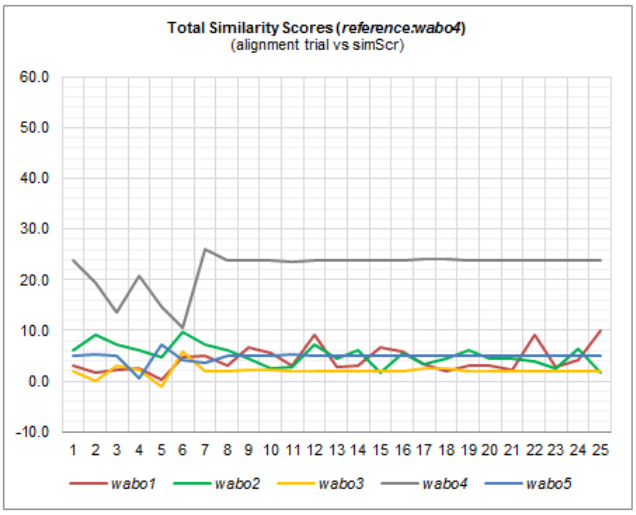


Figure 6.36. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo4*) (*X-axis:alignment runID, Y-axis:similarity score*).

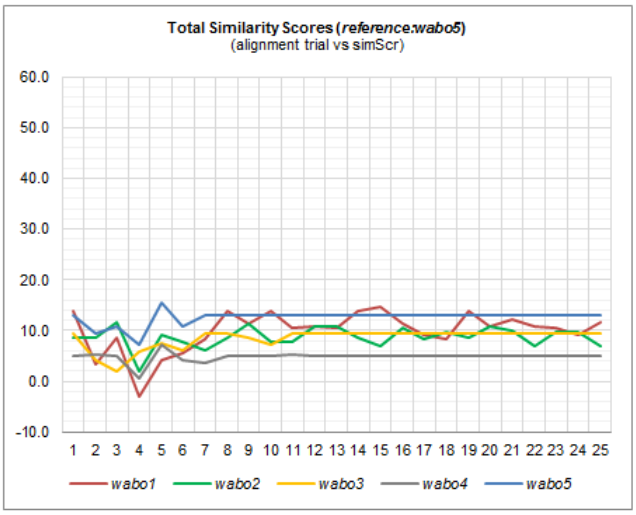


Figure 6.37. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo5*) (*X-axis:alignment runID, Y-axis:similarity score*).

Actually the corresponding similarity scores (*simScr*) are significantly dependent to the runtime configurations for each alignment run, which are:

- Confidence threshold (*confThr*) determined at the underlying alignment run
- Composition of the sequence (i.e. the order of tasks) figured out by the dominant behavior for source and target individuals
- The corresponding confidence table valid for the underlying process alternative and alignment run

There emerges a requirement for the preprocessing step to normalize biased similarity scores and convert these values to proper distance attributes before process clustering. As stated in [92], cosine similarity is an appropriate measurement to transform the similarity scores and it holds the properties of distance metric, i.e. non-negativity, symmetry, zero-value and triangle inequality properties. Hence, the cosine of the angle between two Eucledian vectors is not affected by the scalar transformation. In this aspect as stated in section 5.2.3, each reference process alternative *i* is converted to a *process variant vector* pv_i^k for each alignment run *k*, and *term weight* (pv_{ij}^k) constitutes the similarity score (*simScr*) between the underlying process alternatives *i* and *j* at alignment run *k*.

Then cosine similarity between the process variant vectors pv_i^k and pv_j^k is measured as the angle between these two n-dimensional process vectors according to Equation 5.11. The value of cosine similarity ranges from -1 (quite distinct) to +1 (equivalent). Following the cosine similarity transformation, normalized similarity values should be converted into distance metrics, i.e. $dist(pv_i,pv_j)$. Indeed, natural transformation such as $dist(pv_i,pv_j) = 1- cosSim(pv_i,pv_j)$ does not guarantee the triangle inequality. As stated in [92], the transformation $\mathrm{dist}(\mathrm{pv_i},\mathrm{pv_j})=\sqrt{1\text{-}\mathrm{cosSim}(\mathrm{pv_i},\mathrm{pv_j})^2}$ is applied to produce distance metric. As a result, each process alternative at an alignment run is converted to an *instance* with 5 numerical (*dist_waboi*) and 2 categorical type (i.e. *runID* and *reference* as *target class*) attributes. Totally 125 instances (25 alignment run x 5 process alternatives) are obtained for Environmental Permit Application use case prior to process clustering step. Figure 6.39 exemplifies the preprocessing step for alignment run 17. Details about cosine similarity transformation for Environmental Permit Application use case are given in Appendix E1.

Indeed, business process clustering is a practical concept to reengineer the current process models or to extract the major commonalities among the process candidates in order to support new process designs. While hierarchical clustering (agglomerative or divisive) is applied in [93, 94], IR-based multimodal search, DBSCAN and k-Means clustering algorithms are also preferred in [94, 95, 96]. In conformance with the prior studies [15, 27, 28] handling Environmental Permit Application use case, the number of clusters (*numbCluster*) is set as 2 and 3 respectively and *Expectation Maximization* (EM), *Hierarchical Clustering* (HC) and *Simple K-Means* are applied with various distance functions (e.g. *manhattan*, *eucledian* and *minkowski* distance functions) as clustering algorithms. Table 6.19 summarizes the clustering results.

According to the clustering content and the number of incorrectly clustered instances with 3 clusters (*numbCluster=3*), all three clustering algorithms return with the exact outcome: while process alternatives *wabo2, wabo3 and wabo5* are grouped in the same cluster, process alternative *wabo1* and *wabo4* are held at distinct clusters. Respectively, Expectation Maximization and Simple K-Means algorithms have a better accuracy than Hierarchical Clustering. As the natural effect of the increase at the cluster number, sum of the within distance at Simple K-Means algorithm diminishes by an average value of 35%. A similar result is also emphasized in [94] such that, K-Means algorithm does not progress the clustering steps upon the prior clustering instances. Hence it results better in clustering in terms of intra- and inter-cluster distance metrics than gained with hierarchical algorithm. Additionally, the log likelihood value of 2.102 for Expectation Maximization application with 3 clusters signals for a better fit to the testing data and it is proposed to choose the model with the largest log likelihood value for local maxima [74].

As given in Figure 6.38, the instance plots, which visualizes each distinct instance according to the distance attributes (*dist_waboi*), clarifies the clustering mechanism such that, the distinction at *dist_wabo4* attribute (i.e. peak values at *dist_wabo4* column for both plots) signifies the singleton-type cluster for process alternative *wabo4*. This dissociation (or segregation) is partially viable for process alternative *wabo1*. Relatively high average values for *dist_wabo1* attribute (i.e. 0.651, 0.611 and 0.596 for *wabo2*, *wabo3* and *wabo5* respectively) hinder any convergence between *wabo1* and *cluster2* {*wabo2, wabo3, wabo5*}.

The second instance plot handles the intra-cluster distance (or cohesion) for *cluster2*. The average distance of 0.469 between process alternatives *wabo2* and *wabo3* highlights a relatively significant commonality between corresponding business context. This neighborhood between these two process variants can be evaluated as an early convergence at clustering iterations. Additionally, the bold thick red and orange lines at the second instance plot also emphasize such an alternative neighborhood between process alternatives *wabo3* and *wabo5* with an average distance of 0.426.

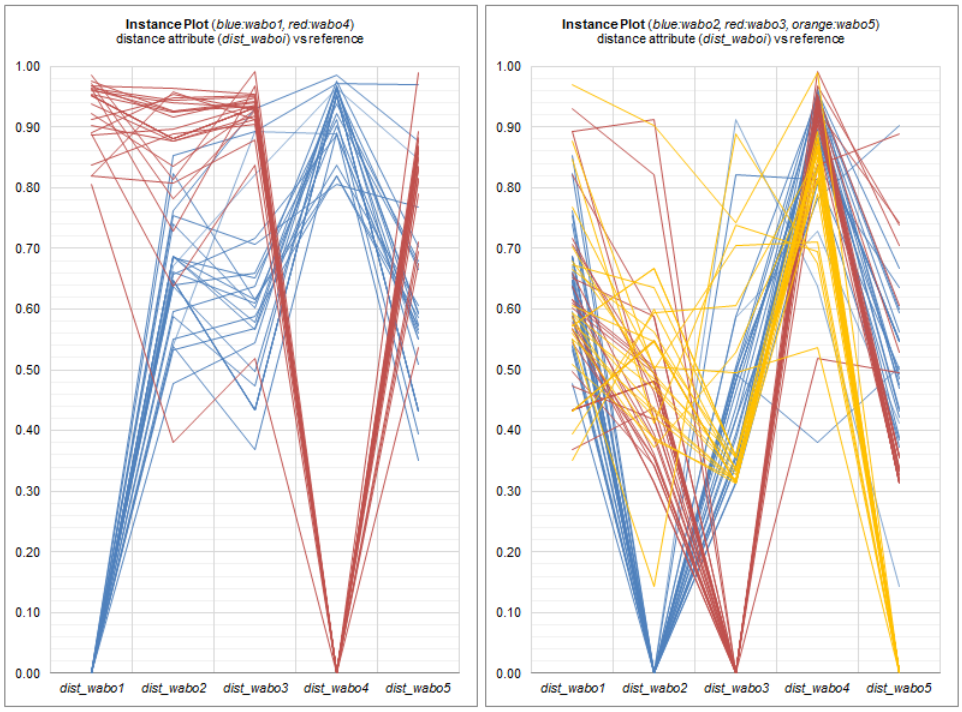


Figure 6.38. Instance Plots for {*wabo1*,*wabo4*} and {wabo2, wabo3, wabo5} Process Clusters for Environmental Permit Application Use Case (*X-axis:distance attributeID, Y-axis:distance value*).

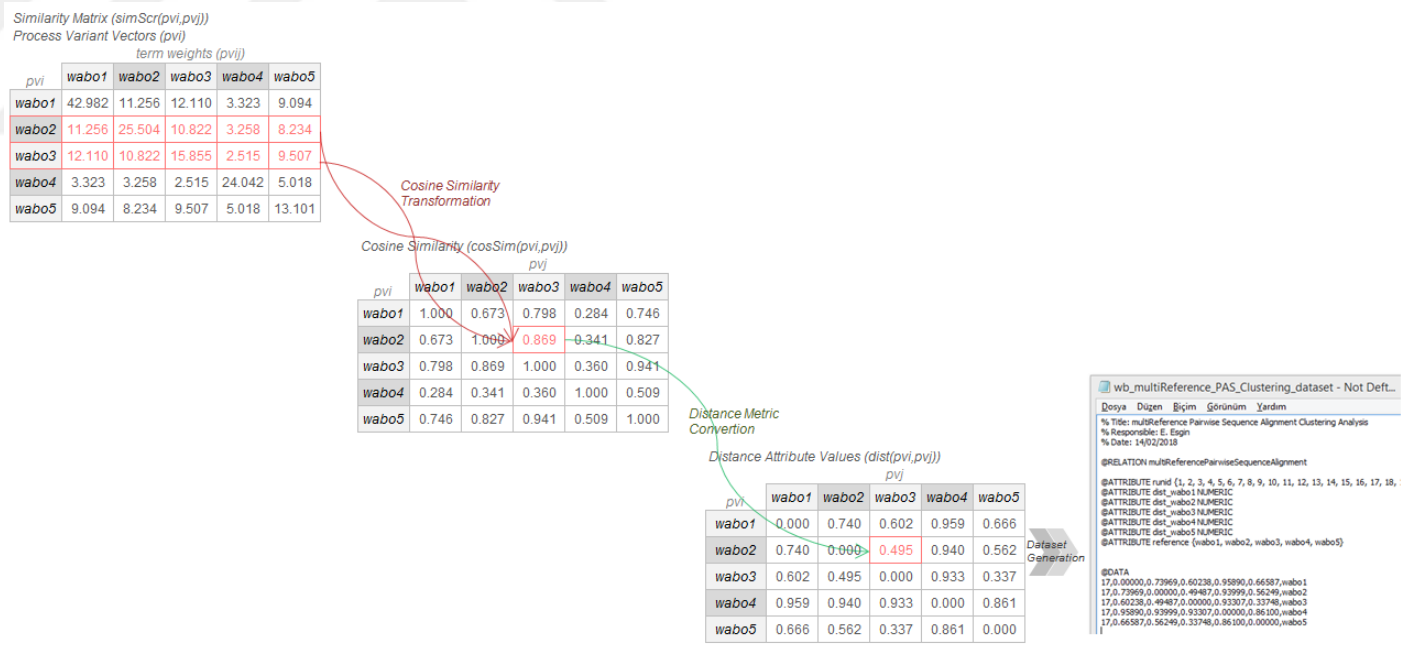


Figure 6.39. Example Preprocessing Step at Alignment Run 17 for Environmental Permit Application Use Case. Preprocessing step consists of two operations: *cosine similarity transformation* and *distance metric conversion*. As a result, each process alternative at a single alignment run is turned into an *instance* consisting of 5 numeric distance attributes and 2 categorical attributes (i.e. *runID* and *reference* as *target class*).

Table 6.19. Process Clustering Results for Environmental Permit Application Use Case. According to the clustering content and the number of incorrectly clustered instances with 3 clusters (*numbCluster*), all three clustering algorithms return with the exact outcome: while process alternatives *wabo2, wabo3 and wabo5* are grouped in the same cluster, process alternative *wabo1* and *wabo4* are held at distinct clusters.

numb Cluster	Expectation Maximization	Hierarchial Clustering			SimpleKMeans	
		Manhattan	Eucledian	Minkovski (order=3)	Manhattan	Eucledian
2	Log likelihood: 0.71418 Process time: 0.01 sec. c0 c1 25 0 wabo1 25 0 wabo2 25 0 wabo3 0 25 wabo4 25 0 wabo5	Incorectly clustered instance: 1 Process time: 0.03 sec. c0 c1 25 0 wabo1 25 0 wabo2 24 1 wabo3 0 25 wabo4 25 0 wabo5	Incorectly clustered instance: 1 Process time: 0.02 sec. c0 c1 25 0 wabo1 24 1 wabo2 25 0 wabo3 0 25 wabo4 25 0 wabo5	Incorectly clustered instance: 1 Process time: 0.03 sec. c0 c1 25 0 wabo1 24 1 wabo2 25 0 wabo3 0 25 wabo4 25 0 wabo5	Incorectly clustered instance: 0 Sum of within cluster dist: 99.1718 c0 c1 0 25 wabo1 0 25 wabo2 0 25 wabo3 25 0 wabo4 0 25 wabo5	Incorectly clustered instance: 0 Sum of within cluster dist: 32.5398 c0 c1 0 25 wabo1 0 25 wabo2 0 25 wabo3 25 0 wabo4 0 25 wabo5
3	Log likelihood: 2.10241 Process time: 0.01 sec. c0 c1 c2 0 0 25 wabo1 25 0 0 wabo2 25 0 0 wabo3 0 25 0 wabo4 25 0 0 wabo5	Incorectly clustered instance: 1 Process time: 0.02 sec. c0 c1 c2 25 0 0 wabo1 0 25 0 wabo2 0 24 1 wabo3 0 0 25 wabo4 0 25 0 wabo5	Incorectly clustered instance: 1 Process time: 0.02 sec. c0 c1 c2 25 0 0 wabo1 0 24 1 wabo2 0 25 0 wabo3 0 0 25 wabo4 0 25 0 wabo5	Incorectly clustered instance: 1 Process time: 0.03 sec. c0 c1 c2 25 0 0 wabo1 0 24 1 wabo2 0 25 0 wabo3 0 0 25 wabo4 0 25 0 wabo5	Incorectly clustered instance: 0 Sum of within cluster dist: 73.2873 c0 c1 c2 0 25 0 wabo1 0 0 25 wabo2 0 0 25 wabo3 25 0 0 wabo4 0 0 25 wabo5	Incorectly clustered instance: 0 Sum of within cluster dist: 19.3989 c0 c1 c2 0 25 0 wabo1 0 0 25 wabo2 0 0 25 wabo3 25 0 0 wabo4 0 0 25 wabo5

Likewise in Environmental Permit Application use case, Multi-Reference Pairwise Alignment comes up with 5 similarity score graphs for Period-End Closing use case given in Figures 6.40 - 6.44. As shown in Figures 6.41 and 6.43, although there happens a solid discrepancy between process alternatives *client2* and *client4*, the lack of some Product Costing and Material Ledger functionalities may result in a posterior neighborhood for these process alternatives. Alternatively, there is a significant correlation between process alternatives *client1* and *client3*, which shows similar process behavior and responses to the runtime configurations that are characterized by process discovery and GA parameters. The secondary positioning of these process alternatives in Figures 6.40 and 6.42 strengthens this outcome, while the *primary position* is always addressed to the reference itself. Respectively, the relative position of *client5* shown in Figure 6.44 implies a posterior grouping with *client1* and *client3* at the following clustering iterations.

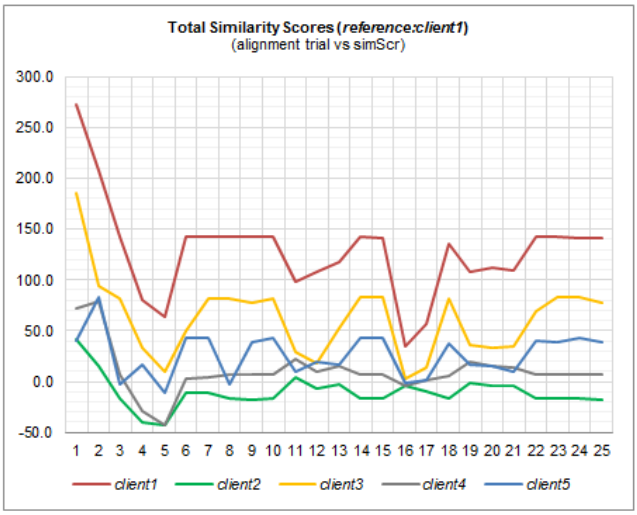


Figure 6.40. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Period-End Closing Use case (*reference:client1*) (*X-axis:alignment runID, Y-axis:similarity score*).

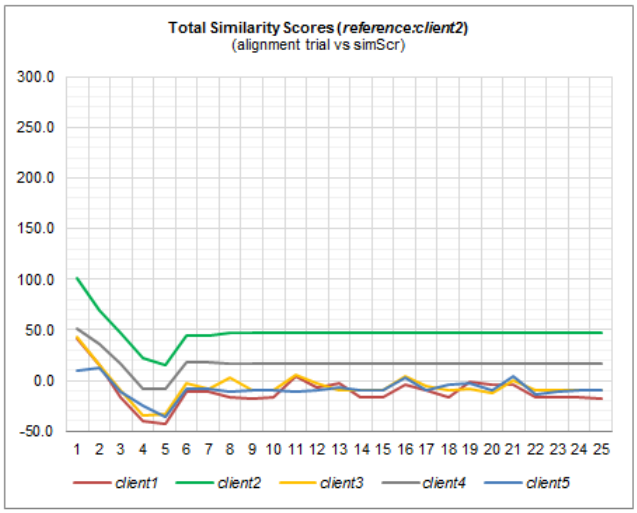


Figure 6.41. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Period-End Closing Use case (*reference:client2*) (*X-axis:alignment runID, Y-axis:similarity score*).

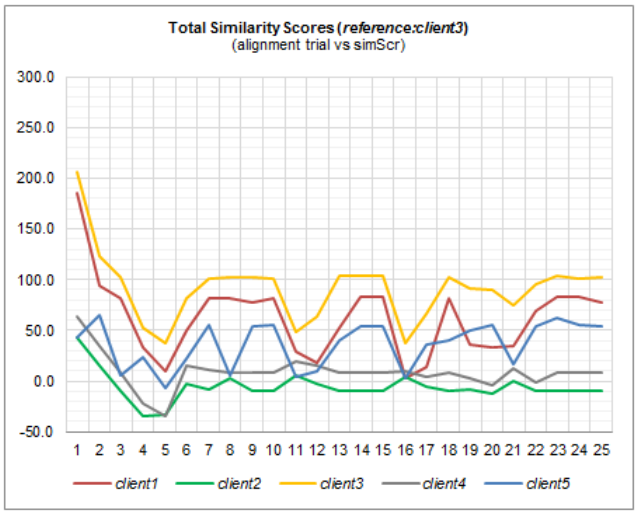


Figure 6.42. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Period-End Closing Use case (*reference:client3*) (*X-axis:alignment runID, Y-axis:similarity score*).

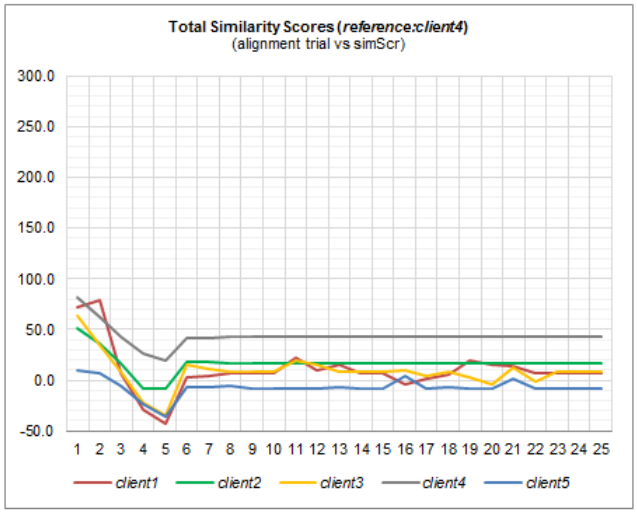


Figure 6.43. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Period-End Closing Use case (*reference:client4*) (*X-axis:alignment runID, Y-axis:similarity score*).

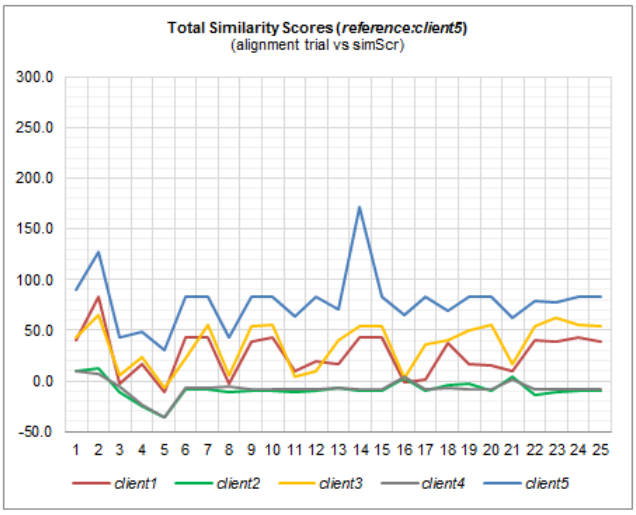


Figure 6.44. Total Similarity Score (*simScr*) per Pairwise Alignment Run for Period-End Closing Use Case (*reference:client5*) (*X-axis:alignment runID, Y-axis:similarity score*).

As stated above, the similarity scores cannot be used as a direct indicator for process clustering, since these values are biased towards alignment runtime configurations, the context of the sequences representing dominant behavior and the activity interactions held at confidence table. Therefore the similarity scores should be normalized and converted into the distance attributes (i.e. *dist_clienti*).

According to [92, 93], cosine similarity is an appropriate metric to normalize the similarity scores such that, each process alternative is turned into process variant vector (pv_i^k) in which term weights correspond the similarity scores. As a following step, the transformation $\mathrm{dist}(\mathrm{pv_i},\mathrm{pv_j})=\sqrt{1/2\left(1\text{-}\cos\mathrm{Sim}(\mathrm{pv_i},\mathrm{pv_j})\right)}$ given in [92] is applied to convert the cosine similarity into distance metric. Consequently, each process variant vector is turned into an instance with 5 numerical (e.g. *dist_clienti*) and 2 categorical attributes (i.e. *runID* and *reference* as *target class*). Figure 6.46 exemplifies the underlying similarity score transformation steps at Period-End Closing use case and details about cosine similarity transformation are given in Appendix E1.

Afterwards, process alternatives are grouped according to the distance attributes (*dist_clienti*) to determine the process families. The results of process clustering can be used to derive generic process models by analyzing common patterns in each process family [93]. Indeed, clustering real-life business processes with respect to *business category* is also performed in [49]. Respectively, the number of clusters (*numbCluster*) is determined as 2 due to implicitly valid industry categories for Period-End Closing use case (i.e. manufacturing and service industries). Clustering run with *numbCluster=3* setting is also performed to analyze the prior clustering iterations. According to this *numbCluster* settings, *Expectation Maximization* (EM), *Hierarchical Clustering* (HC) and *Simple K-Means* clustering algorithms are applied with various distance functions (e.g. *manhattan*, *eucledian* and *minkowski* distance functions) as shown in Table 6.20.

Clustering runs with *numbCluster=2* setting have exactly the same outcome: while process alternatives *client1*, *client3* and *client5* are grouped in the same cluster, process alternatives *client2* and *client4* are assigned to the other cluster. According to the number of incorrectly clustered instances, simple K-Means and EM clustering algorithms have a better accuracy than hierarchical clustering. This is potentially due to the myopic clustering strategy of hierarchical clustering which depends on the previously found sub-clusters. On the other hand, EM and simple K-Means are randomized algorithms and their runs are undeterministic, i.e. possibly resulting in several different clustering runs for the same data set and the number of clusters [94]. In the case of clustering run with *numbCluster=3* setting, the newly created cluster is useless for HC (*cluster0*) such that, no appropriate label can be assigned to this cluster. Unlikely, EM prefers to assign process alternative *client5* to the new cluster (*cluster1*), and simple K-Means prefers to detach the prior cluster {*client2,client4*} into two singleton clusters.

According to the instance plots given in Figure 6.45, the peak values concentrated at around [0.65, 0.85] interval highlights the significant segregation between {*client2, client4*} and {*client1, client3, client5*}. The average distance of value 0.449 between the process alternatives *client2* and *client4* implies a loose cohesion between the corresponding business contexts. Hence this may refer to a late convergence for these process variants. On the other hand, the thickness of red and blue lines at *dist_client1* and *dist_client3* attributes at the second instance plot (with an average distance of 0.27) emphasizes a relatively stronger commonality between the process alternatives *client1* and *client3*. Correspondingly, process alternative *cilent5* converges to process cluster {*client1, client3*} at the later clustering iterations.

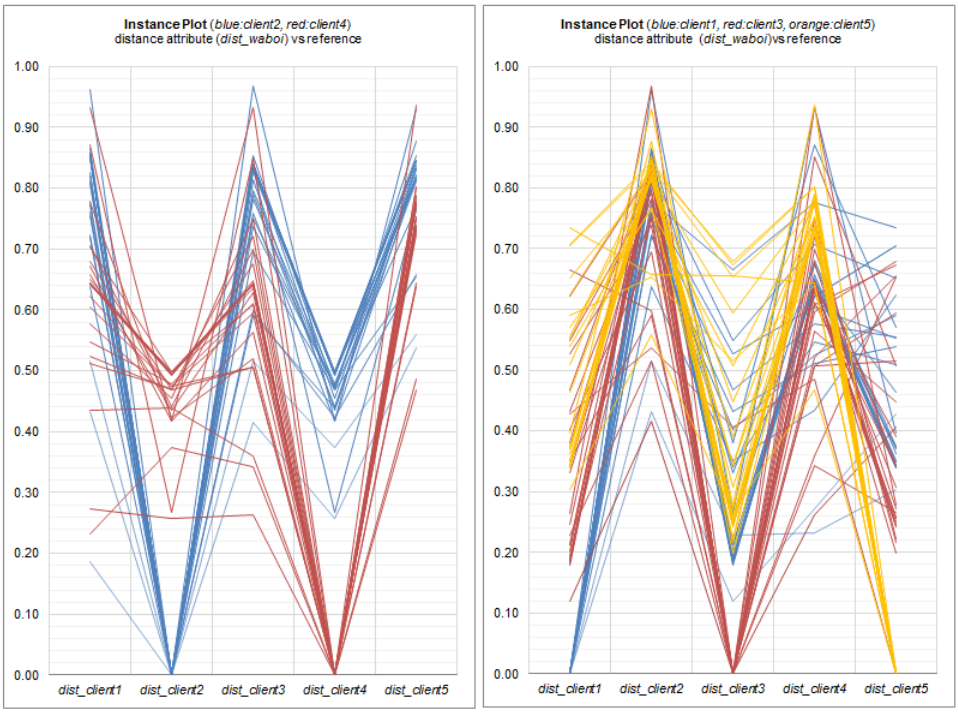


Figure 6.45. Instance Plots for {*client2*, *client4*} and {*client1, client3, client5*} for Period-End Closing Use Case (*X-axis:distance attributeID, Y-axis:distance value*).

Details about Multi-Reference Pairwise Alignment for Environmental Permit Application and Period-End Closing use cases (i.e. aligned forms of dominant behavior sequences per reference selection and similarity scores) are given in Appendix E2.

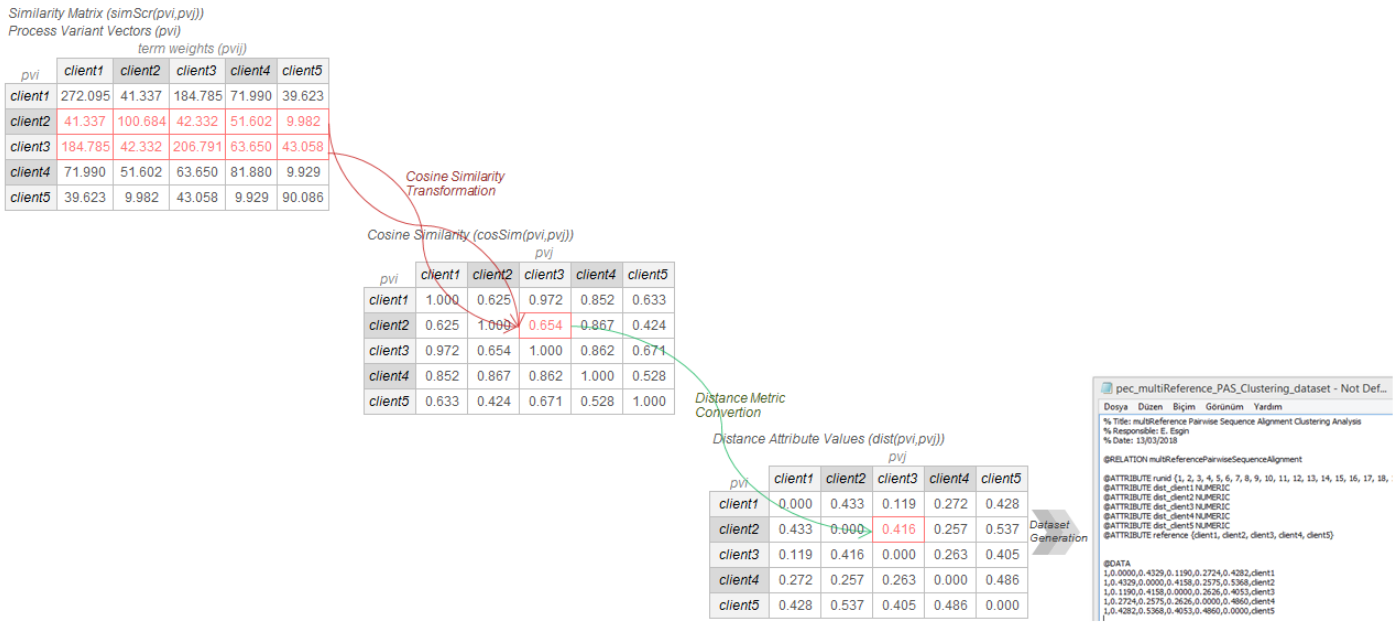


Figure 6.46. Example Preprocessing Step at Alignment Run 1 for Period-End Closing Use Case. Preprocessing step consists of two operations: *cosine similarity transformation* and *distance metric conversion*. As a result, each process alternative at a single alignment run is turned into an *instance* consisting of *n* numeric distance attributes and 2 categorical attributes (i.e. *runID* and *reference* as *target class*).

Table 6.20. Process Clustering Results for Period-End Closing Use Case. While process alternatives *client1*, *client3* and *client5* are grouped in the same cluster, process alternatives *client2* and *client4* are assigned to the other cluster. According to the number of incorrectly clustered instances, simple K-Means and EM clustering algorithms have a better accuracy than HC.

numb Cluster	Expectation Maximization	Hierarchial Clustering			SimpleKMeans	
		Manhattan	Eucledian	Minkovski (order=3)	Manhattan	Eucledian
2	Log likelihood: 1.00371 Process time: 0.05 sec. c0 c1 25 0 client1 0 25 client2 25 0 client3 0 25 client4 25 0 client5	Incorectly clustered instance: 2 Process time: 0.09 sec. c0 c1 25 0 client1 0 25 client2 25 0 client3 2 23 client4 25 0 client5	Incorectly clustered instance: 2 Process time: 0.03 sec. c0 c1 25 0 client1 0 25 client2 25 0 client3 2 23 client4 25 0 client5	Incorectly clustered instance: 2 Process time: 0.06 sec. c0 c1 25 0 client1 0 25 client2 25 0 client3 2 23 client4 25 0 client5	Incorectly clustered instance: 0 Sum of within cluster dist: 97.7788 c0 c1 0 25 client1 25 0 client2 0 25 client3 23 2 client4 0 25 client5	Incorectly clustered instance: 0 Sum of within cluster dist: 24.2447 c0 c1 0 25 client1 25 0 client2 0 25 client3 23 2 client4 0 25 client5
3	Log likelihood: 1.768 Process time: 0.04 sec. c0 c1 c2 0 0 25 client1 25 0 0 client2 0 0 25 client3 25 0 0 client4 0 23 2 client5	Incorectly clustered instance: 6 Process time: 0.02 sec. c0 c1 c2 2 0 23 client1 0 25 0 client2 2 0 23 client3 2 23 0 client4 0 0 25 client5	Incorectly clustered instance: 6 Process time: 0.02 sec. c0 c1 c2 2 0 23 client1 0 25 0 client2 2 0 23 client3 2 23 0 client4 0 0 25 client5	Incorectly clustered instance: 1 Process time: 0.04 sec. c0 c1 c2 25 0 0 client1 0 24 1 client2 25 0 0 client3 2 23 0 client4 25 0 0 client5	Incorectly clustered instance: 0 Sum of within cluster dist: 72.1959 c0 c1 c2 0 25 0 client1 0 0 25 client2 0 25 0 client3 25 0 0 client4 0 25 0 client5	Incorectly clustered instance: 1 Sum of within cluster dist: 18.023 c0 c1 c2 0 25 0 client1 1 0 24 client2 0 25 0 client3 25 0 0 client4 0 25 0 client5

6.4.3. Multi-Sequence Alignment Based Analysis

6.4.3.1. Confidence Enhanced Sequence Alignment Analysis

As stated in section 5.2.2, Multi-Sequence Alignment is a progressive alignment technique that utilizes a confidence enhanced costing function based on Equations 5.4 - 5.10 according to the progressive fashion and constructs the process families depicting the commonalities and discrepancies between the corresponding process alternatives. These cluster contents are illustrated by the process family tree which is the hierarchical arrangement of the process clusters.

In order to analyze the effect of confidence enhanced costing function, that dynamically determines the cost of matching or inDel (insertion/deletion) edit operations according to the *confidence values* that reflect the business rules, *alignment mode* parameter is designed at the Confidence-Enhanced Multi-Sequence Alignment application. This parameter has three settings:

- *Confidence Enhanced SA* (*sequence alignment*). Matching and inDel operations are dynamically valuated by the corresponding element, whether it conforms the business rules that are figured out by the confidence values. Respectively, this mode applies the costing function given as Equations 5.4-5.10.
- *Classical NW* (*Needleman Wunsch*). While matching operation is denoted by confidence threshold (+*confThr*) default value, mismatching or inDel operations are penalized by –*confThr* value. In other words, classical NW mode just applies the activity label similarity proposed in [66]. This label similarity between activities is computed from the activity labels using basic atomic syntactical comparison (the same or different).
- *Sum-of-Pairs* (*SP*). SP is one of the most popular scoring mechanisms for the multiple sequence alignment of genomic sequences and it is also applied in process mining. In this technique, score for multi-sequence alignment of N sequences is calculated as the summation of the scores of all $N^{x}(N-1)/2$ ordinary pairwise alignments of each pair of input sequences of the original candidate multi-sequence alignment [21, 62, 84, 97].

The menu bar for alignment mode parameter at Confidence Enhanced Multi-Sequence Alignment application is shown in Figure 6.47. The effect of sum-of-pairs alignment mode is specifically criticized in Section 6.4.3.2.

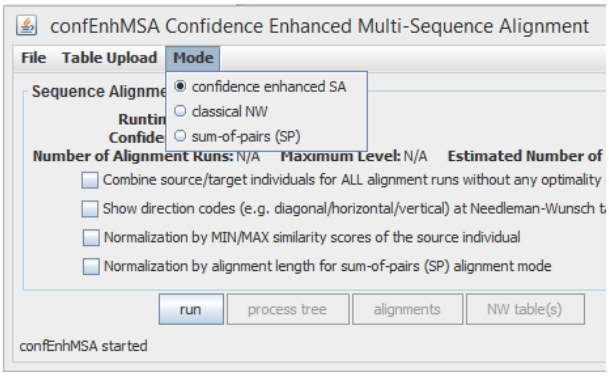


Figure 6.47. Alignment Mode for ConfEnhMSA (Confidence Enhanced Multi-Sequence Alignment) Application.

Multi-Sequence Alignment is distinctly performed for Environmental Permit Application use case with these two alignment modes and process family tree is analyzed at cutting level *level=3* for all 25 alignment runs. The histogram given in Figure 6.48 summarizes the most frequent cluster contents for each alignment mode. Since classical NW mode just handles the activity label similarity and valuates the alignment operations on an atomic similarity scale, *wabo1* and *wabo2*, i.e. the process alternatives modeled by more complex and deeper process maps with higher connectivity, are assigned to distinct singleton clusters for 18 and 14 alignment runs respectively. Moreover, {*wabo3, wabo4, wabo5*}, {*wabo3, wabo5*}, {*wabo4*} and {*wabo1, wabo2*} are other frequent cluster contents. On the other hand, confidence enhanced SA mode assigns process alternative *wabo4* to a singleton cluster for 22 runs.

{*wabo2, wabo3*}, {*wabo1, wabo5*}, {*wabo1, wabo3, wabo5*} and {*wabo1*} constitute other frequent cluster contents. This outcome is respectively in parallel with Multi-Reference Pairwise Alignment. Figure 6.49 summarizes the range of process family tree topologies obtained throughout all alignment runs. Figures 6.50 and 6.51 depict the most frequent process family tree instances per alignment mode.

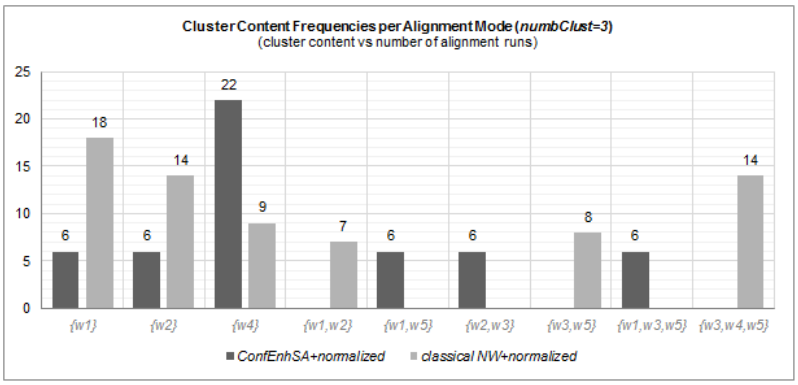


Figure 6.48. Cluster Content Frequencies per Alignment Mode for Environmental Permit Application Use Case.

As stated in section 6.4.2, the similarity scores are transformed into cosine similarity, in which $cosSim(wabo_i,wabo_j)$=1.0 implies a perfect match between the input process alternatives, $wabo_i$ and $wabo_j$. According to Figure 6.52, when the alignment runs that are handled in detailed at the alignment matrices given in Tables 6.27 and 6.28 are focused, it is realized that *cosSim*(*wabo2,wabo3*) outperforms the *average cosine similarity*. Hence this strong commonality enforces the process alternatives *wabo2* and *wabo3* to instantiate the first cluster *cluster0* at *level=2* as shown in Figure 6.50. Due to the increase at the distance between the centroid of *cluster0* and the instances belonged to *wabo1* and *wabo5*, process alternatives *wabo1* and *wabo5* are merged and instantiate the second cluster *cluster1* at the next level. Actually the outlier-like behavior of process alternative *wabo4* also affects these segregations at Multi-Sequence Alignment approach with confidence enhanced SA mode.

Alternatively in order to interpret the mechanism of confidence enhanced cost functioning, the *length* and *similarity score distributions* are also analyzed for the *root node* of process family trees given in Figures 6.50 and 6.51. Since matching and inDel edit operations are scored by +/-*confThr* default values, total similarity score for Multi-Sequence Alignment with classical NW mode is staggered at [-4, -1] interval as shown in Figure 6.53.

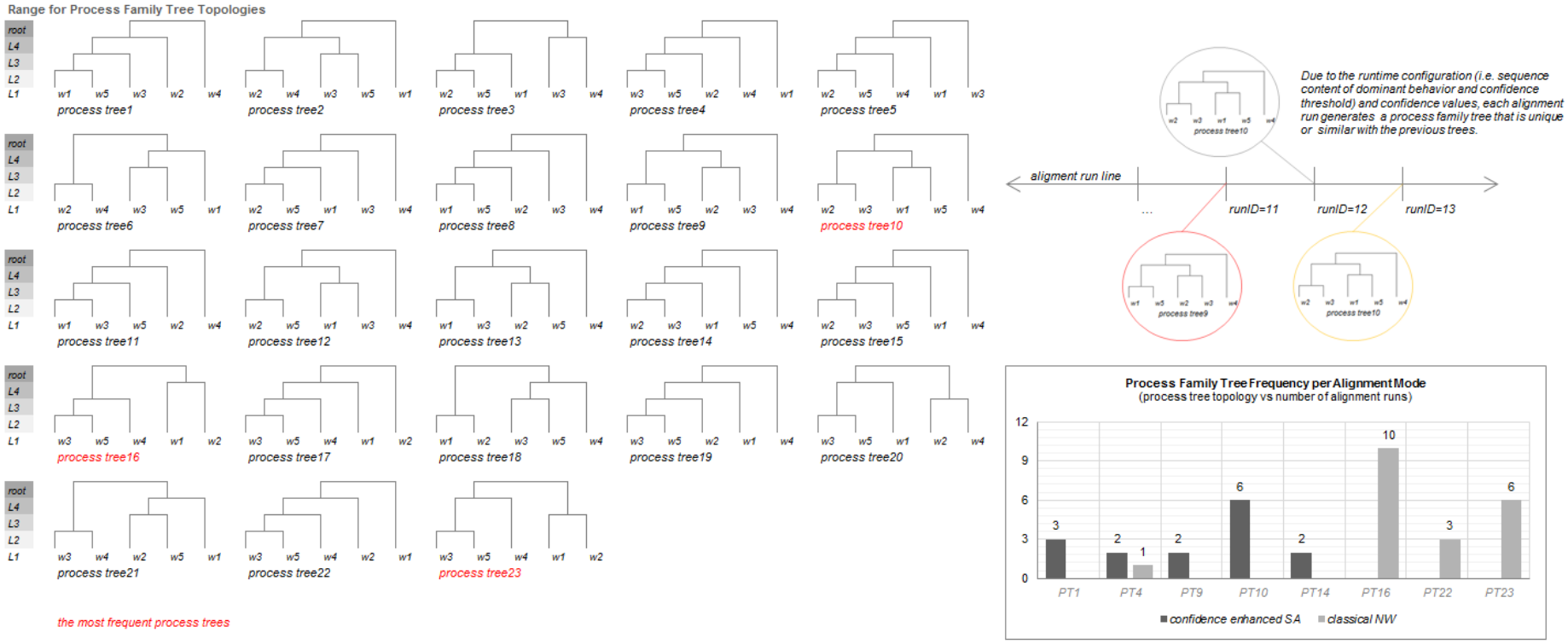


Figure 6.49. Range for Process Family Tree Topologies and Process Family Tree Frequency per Alignment Mode for Environmental Permit Application Use Case. Due to the runtime configuration (i.e. sequence content of dominant behavior and confidence threshold) and confidence values, each alignment run generates a process family tree that is unique or similar with the previous trees. Respectively, *PT10* for confidence enhanced SA and *PT16* for classical NW mode are the most frequent process family tree topologies.

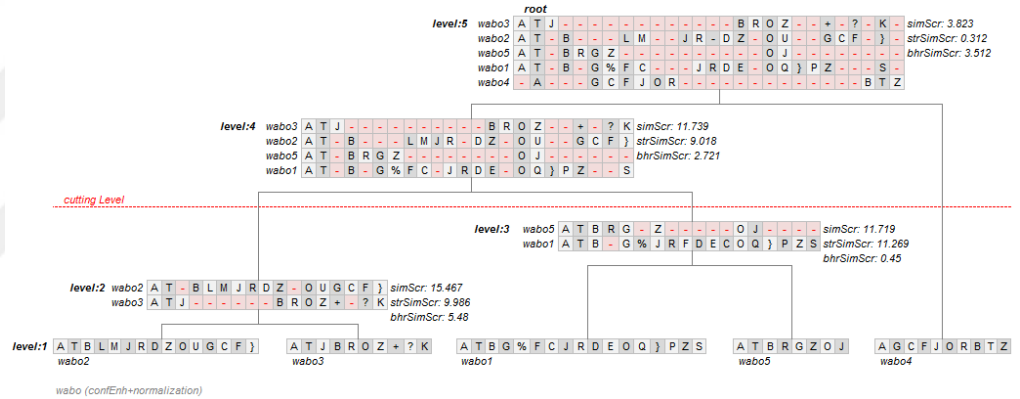


Figure 6.50. Process Family Tree Instance for Multi-Sequence Alignment with Confidence Enhanced SA Mode at Environmental Permit Application Use case. At the cutting level (*level=3*), the clusters {*wabo2, wabo3*}, {*wabo1, wabo5*} *and* {*wabo4*} are instantiated. The underlying process tree topology is shown as *PT10* in Figure 6.49.

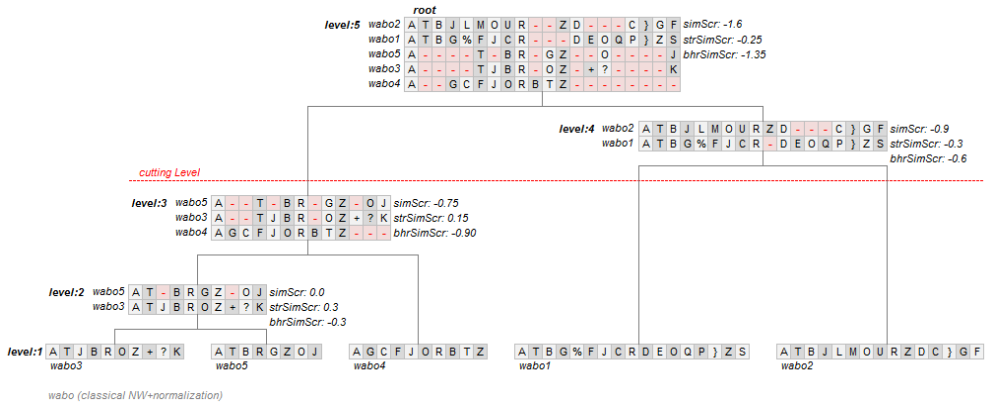


Figure 6.51. Process Family Tree Instance for Multi-Sequence Alignment with Classical NW Mode at Environmental Permit Application Use case. At the cutting level (*level=3*), the clusters {*wabo3, wabo4, wabo5*}, {*wabo1*} *and* {*wabo2*} are instantiated. The underlying process tree topology is shown as *PT16* in Figure 6.49.

Since confidence threshold and the content of dominant behavior sequences are slightly altered throughout the alignment runs, there happens a steady-state phase after alignment run 7 for classical NW mode. On the other hand, confidence enhanced SA mode implies a more sensitive alignment costing with respect to significant fluctuations at similarity values as shown in Figure 6.53.

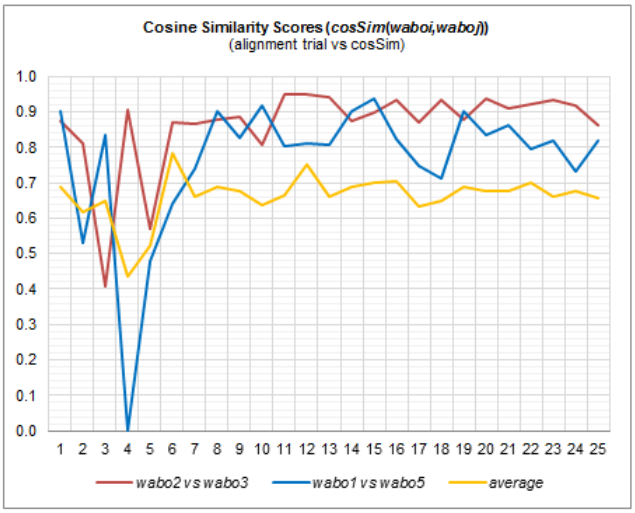


Figure 6.52. Cosine Similarity Scores for Process Families {*wabo2, wabo3*} and {*wabo1, wabo5*} at Environmental Permit Application Use Case (*X-axis:alignment runID, Y-axis:cosine similarity score*).

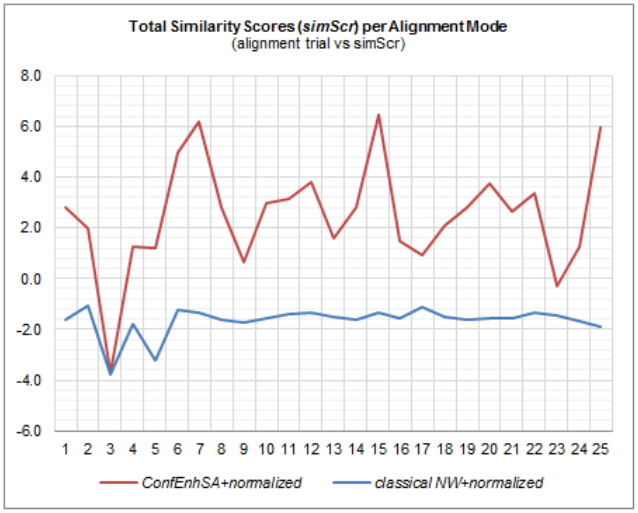


Figure 6.53. Total Similarity Scores per Alignment Mode for Environmental Permit Application Use Case (*X-axis:alignment runID, Y-axis:similarity score*). The confidence-aware dynamic cost functioning feature of confidence enhanced SA mode results in a wider range for similarity scores.

Figure 6.54 highlights the rationale that Multi-Sequence Alignment with classical NW mode prefers the matching rather than the inDel edit operation with an average length of 19 units. Hence classical NW mode is enforced for mismatching and totally penalized due to the negative structural similarity as shown in Figure 6.55. Moreover, since classical NW mode always assigns the gap penalty of *–confThr* to the inDel edit operation, the behavioral similarity scores are almost stabilized at approximately -1.0 level as shown in Figure 6.56. Controversially, positive structural similarity obtained at the alignments with confidence enhanced SA mode highlights the fact that, activity substitutions are encouraged to be replaced according to the substantive business knowledge. While this mode tends to highly penalize the inDel edit operations that contradict with the business context, approximately 52% of all alignment runs have a positive behavioral similarity scores as shown in Figure 6.56.

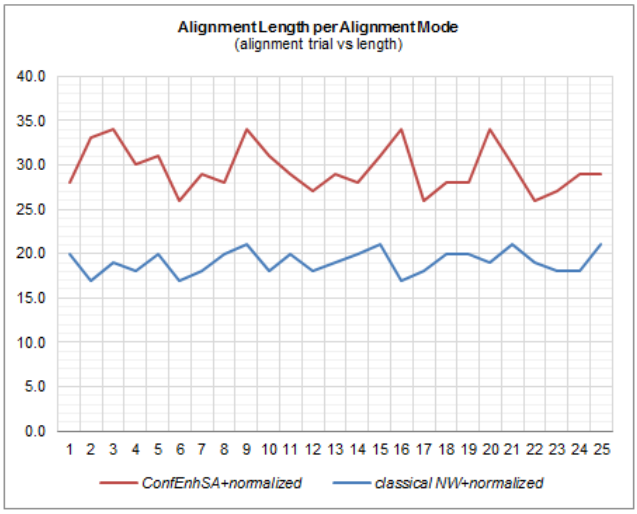


Figure 6.54. Alignment Length per Alignment Mode for Environmental Permit Application Use Case (*X-axis:alignment runID, Y-axis:alignment length*). Classical NW mode has some limitations in reflecting the functional validity by assigning default +/-*confThr* value to matching and inDel operations. Hence it tends to apply matching operations and this implies relatively shorter alignments at the root node of process family tree.

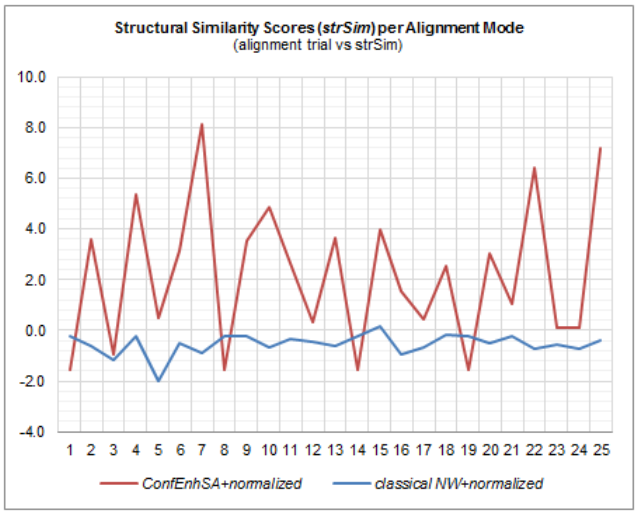


Figure 6.55. Structural Similarity Scores per Alignment Mode for Environmental Permit Application Use Case (*X-axis:alignment runID, Y-axis:structural similarity score*).

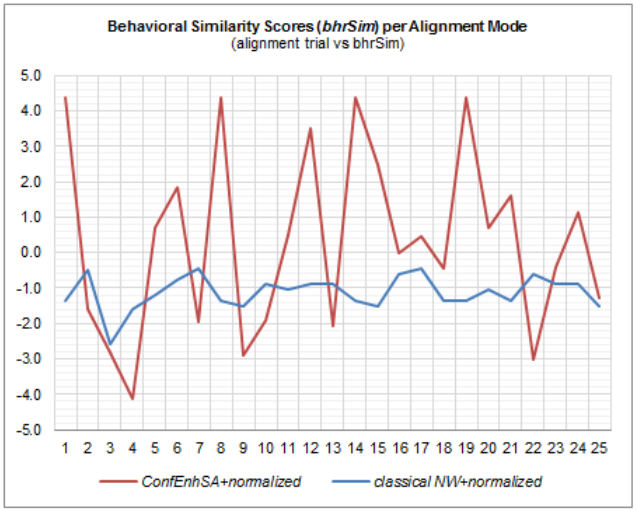


Figure 6.56. Behavioral Similarity Scores per Alignment Mode for Environmental Permit Application Use Case (*X-axis:alignment runID, Y-axis:behavioral similarity score*).

Alternatively, Multi-Sequence Alignment is also distinctly performed for Period-End Closing use case with confidence enhanced SA and classical NW alignment modes. Due to valid business categorization for the corresponding use case (i.e. two distinct business categories as manufacturing and service), number of clusters (*numbCluster*) parameter is set as 2 and the process family tree instances are analyzed at cutting level *level=4* for all 25 alignment runs. Histogram given in Figure 6.57 summarizes the most frequent cluster contents such that, two distinct alignment modes have a consensus on {*client2, client4*} and {*client1, client3, client5*} process clustering. In fact, such an industry-based process clustering highlights the rationale that, industry level business requirements dominate and delimit the software component activations; therefore reference business models and the set of valid activities are determined according to these software components. Additionally, this clustering outcome is consistent with Multi-Reference Pairwise Alignment.

In order to interpret the effect of confidence enhanced costing function and the hierarchical clustering mechanism (e.g. the topology and the branching order at process tree), it is also considered to analyze the cluster contents for *numbCluster=3* setting as shown in Figure 6.58. While classical NW mode prefers to instantiate the clusters {*client2, client4*} and {*client1, client3*}, as an alternative confidence enhanced SA mode initially considers the process cluster {*client1, client3, client5*} with two singleton clusters for process alternatives *client2* and *client4* at *level=3*. This is due to strong distinction between the business context of the corresponding process alternatives in service industry, while confidence enhanced SA mode propagates the merge of process alternatives *client2* and *client4* to later clustering iterations because of weak commonalities between the corresponding business requirements, classical NW mode solely determines the activity label similarities to minimize the gap penalties (or negative behavioral similarities) due to improper inDel operations. Figure 6.60 summarizes the range of process family tree topologies obtained throughout all alignment runs and the frequencies of the corresponding process family trees according to the alignment mode. Figures 6.61 and 6.62 depict the most frequent process family tree instances per alignment mode.

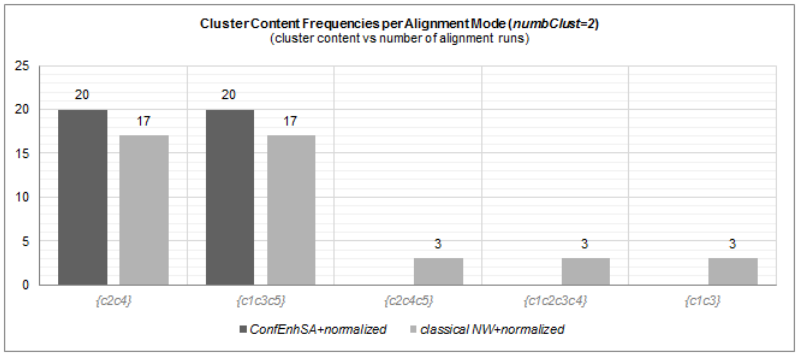


Figure 6.57. Cluster Content Frequencies per Alignment Mode for Period-End Closing Use Case (*numbCluster=2*).

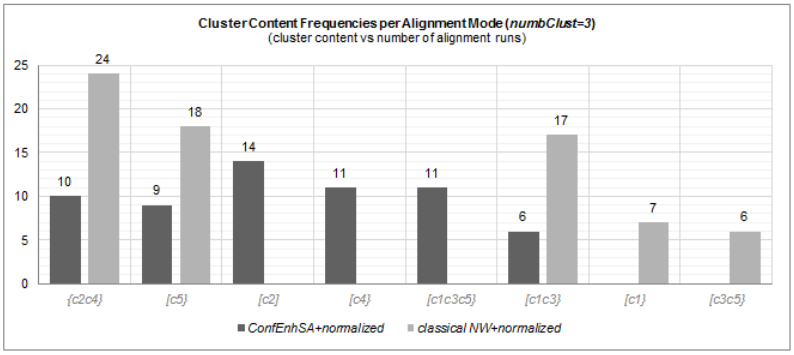


Figure 6.58. Cluster Content Frequencies per Alignment Mode for Period-End Closing Use case (*numbCluster=3*).

Likewise in Environmental Permit Application use case, it is better to visualize the cosine similarity measurements in order to interpret the mechanism of costing function at Multi-Sequence Alignment. These scores are used as the metric to transform the similarity scores that are highly correlated to the confidence threshold and valid confidence table at the corresponding alignment run. According to Figure 6.59, *cosSim*(*client1,client3*) is quite higher than *average cosine similarity* and other candidate pairs. Hence underlying significant relation enforces the process alternatives *client1* and *client3* to instantiate the first cluster *cluster0* at *level=2*. Indeed, this process clustering is consistent according to the variety of SAP/CO components that perfectly match for these two process alternatives. Although the instantiation of *cluster0* increases the average distance between the centroid of *cluster0* and process alternative *client5*, there happens a late convergence of *client5* to *cluster0* at the following level (*level=3*). Finally, process alternatives operating at service industry (i.e. *client2* and *client4*) are merged as *cluster1* at *level=4*.

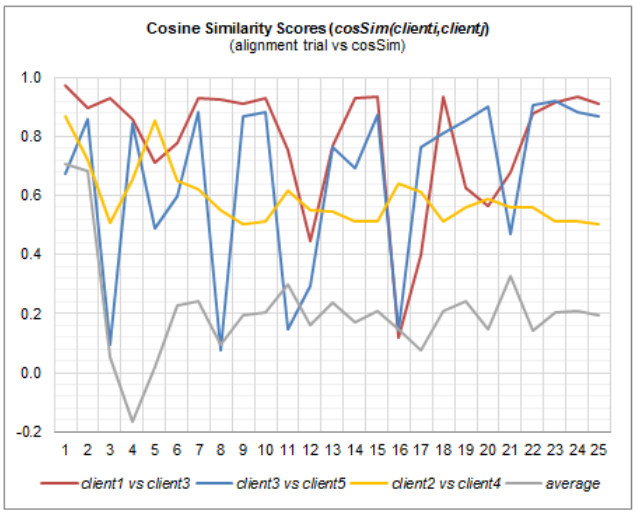


Figure 6.59. Cosine Similarity Scores for Process Families {*client1, client3*}, {*client3, client5*} and {*client2, client4*} at Period-End Closing Use Case (*X-axis:alignment runID, Y-axis:cosine similarity score*).

As an alternative analysis, the characteristics of the *root node* at the process family tree are interpreted to evaluate the effect of business context-aware costing function. As shown in Figure 6.63, both alignment modes are conservative with respect to the matching edit operation such that, confidence enhanced SA mode often prefers the matching rather than inDel operation with an average length of 51 units. This is due to the fact that; there is a significant behavioral discrepancy based on the business context between the manufacturing and service industries. For instance, insertion of product costing (PC) or material ledger (ML) related activities into service-type process alternatives' sequence are discouraged by highly negative inDel costs given in Equations 5.9 and 5.10. Hence the substitution of uncorrelated or contrasting elements is highly penalized, while substitute activities are encouraged to be replaced according to substantive business knowledge. As a result, confidence enhanced SA mode tends to call matching and has relatively higher structural similarity than behavioral similarity as shown in Figures 6.64 and 6.65.

Likewise, classical NW mode is specialized on matching due to the gap penalty assigned by inDel edit operation. Hence while structural similarity scores are damped by indispensable mismatching operations, behavioral similarity is stabilized at approximately -5.0 level, which is relatively better than the behavioral similarity scores obtained by confidence enhanced SA mode. Consequently, total similarity scores of classical NW mode are stabilized at -5.0 level as shown in Figure 6.66.

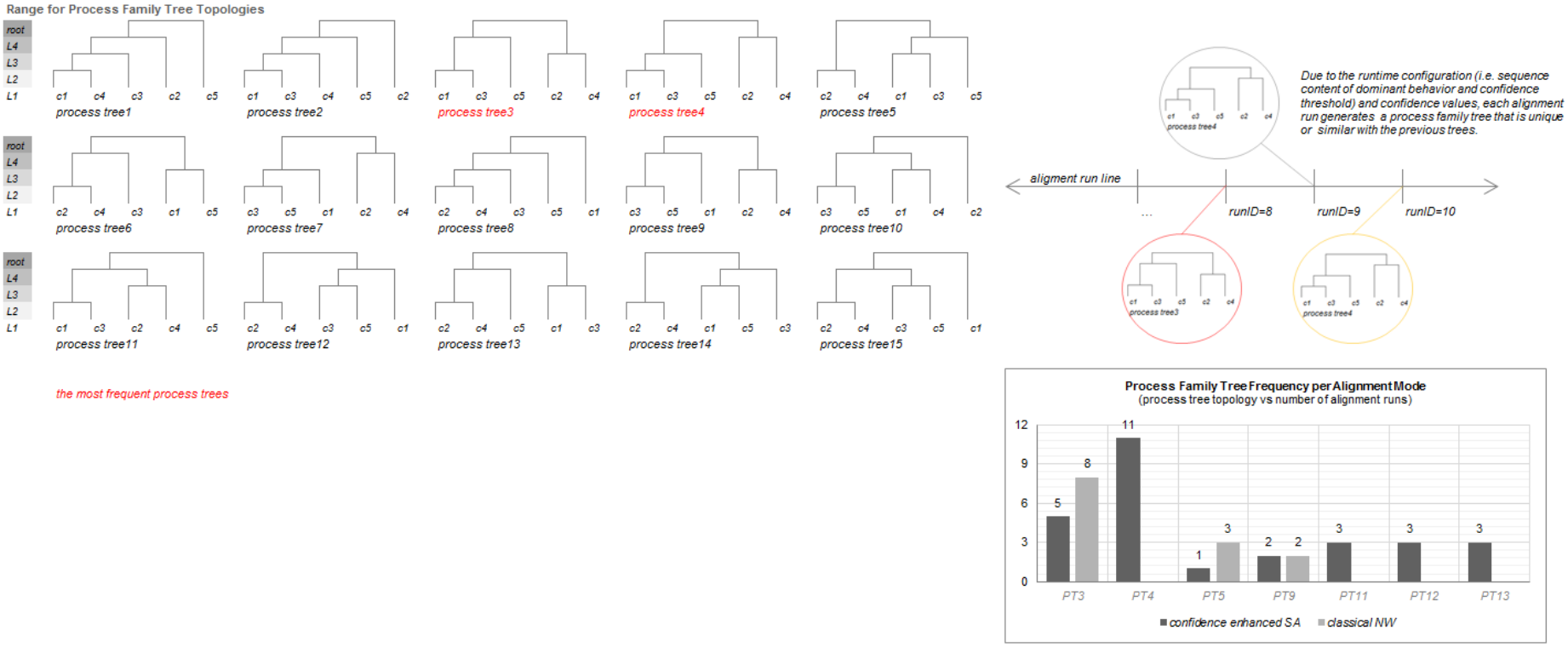


Figure 6.60. Range for Process Family Tree Topologies and Process Family Tree Frequency per Alignment Mode for Period-End Closing Use Case. Respectively, *PT4* for confidence enhanced SA and *PT3* for classical NW mode are the most frequent process family tree topologies. Because of strict discrepancies between the business requirements of the corresponding manufacturing and service industries, the range of process family trees is shrunk with respect to the range in Environmental Permit Application use case given in Figure 6.49.

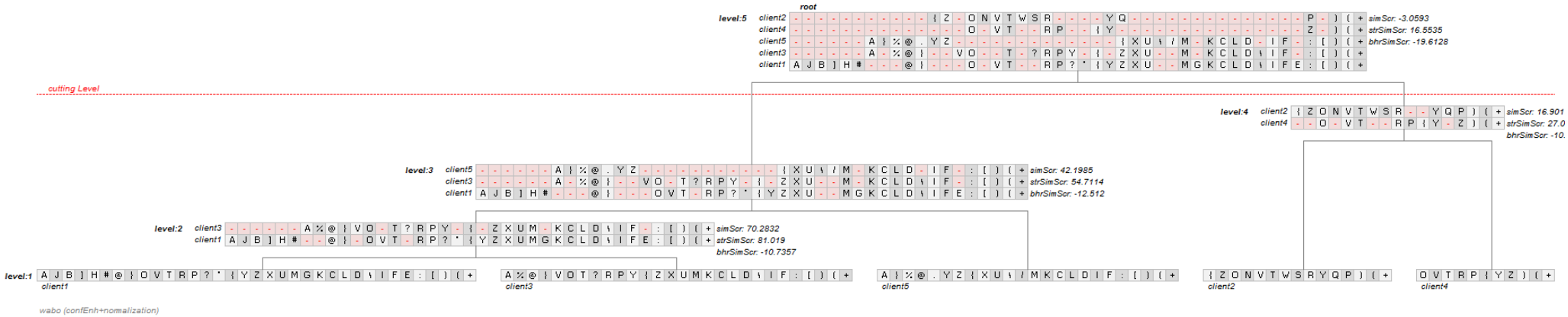


Figure 6.61. Process Family Tree Instance for Multi-Sequence Alignment with Confidence Enhanced SA Mode at Period-End Closing Use Case. At the cutting level (*level=4*), the clusters {*client2, client4*} and {*client1, client3, client5*} are instantiated. The underlying process family tree topology is shown as *PT4* in Figure 6.60.

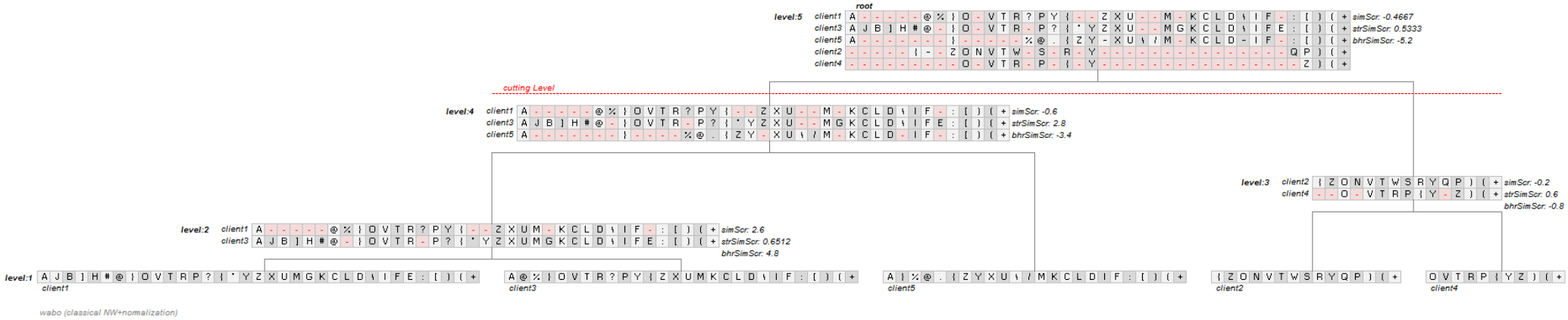


Figure 6.62. Process Family Tree Instance for Multi-Sequence Alignment with Confidence Enhanced SA Mode at Period-End Closing Use Case. At the cutting level (*level=4*), the clusters {*client2, client4*} and {*client1, client3, client5*} are instantiated. The underlying process family tree topology is shown as *PT3* in Figure 6.60.

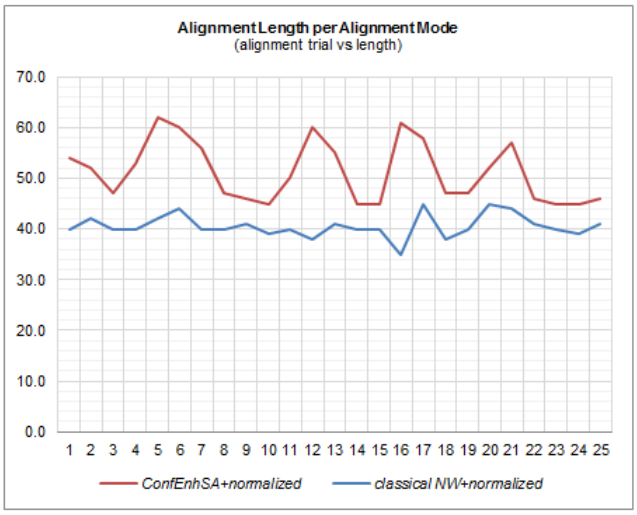


Figure 6.63. Alignment Length per Alignment Mode for Period-End Closing Use Case (*X-axis:alignment runID, Y-axis:alignment length*). Due to the limitations of confidence-enhanced costing function that highly penalizes the insertion and substitution of uncorrelated or contrasting elements, confidence enhanced SA mode prefers the matching operation rather than inDel.

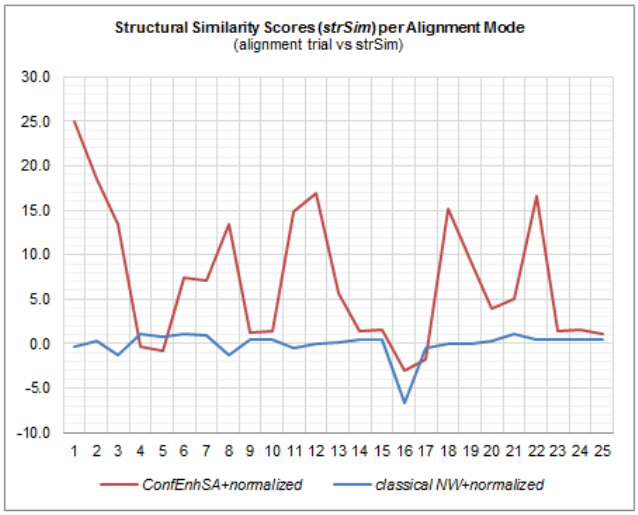


Figure 6.64. Structural Similarity Scores per Alignment Mode for Period-End Closing Use Case (*X-axis:alignment runID, Y-axis:similarity score*).

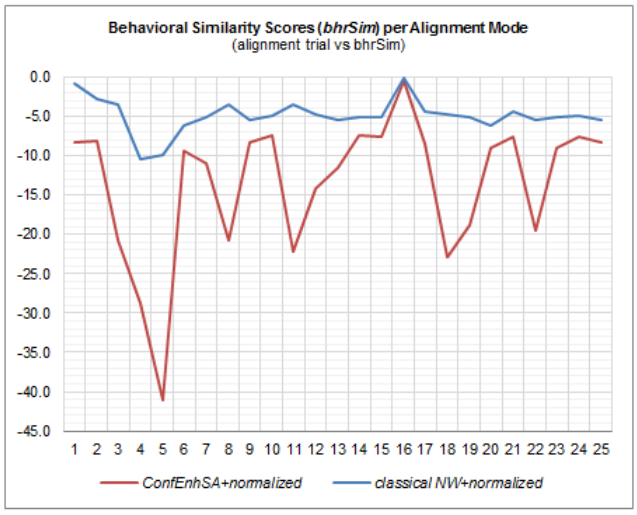


Figure 6.65. Behavioral Similarity Scores per Alignment Mode for Period-End Closing Use Case (*X-axis:alignment runID, Y-axis:behavioral similarity score*).

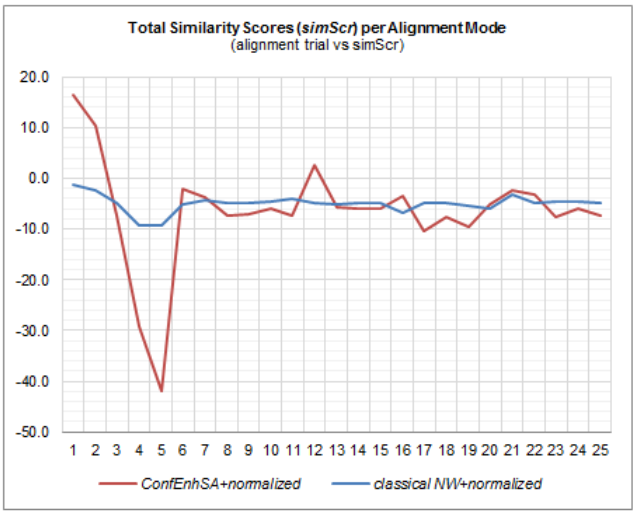


Figure 6.66. Total Similarity Scores per Alignment Mode for Period-End Closing Use Case (*X-axis:alignment runID, Y-axis:similarity score*).

Sample alignment run list and process tree outputs for alignment run 21 at Environmental Permit Application use case are given in Appendix F.

6.4.3.2. Comparison with Prior Cost Functions

As stated in Section 5.2, sequence alignment is a standard technique in bioinformatics domain for visualizing the correlations between the regions in a set of evolutionary-related structures. While the similarity among these structures can be determined by pairwise alignment, there is a fundamental requirement for multi-sequence alignment. Major reason is that pairwise alignment is insufficient to pinpoint the conserved regions among the sequences [22]. Adapting the definition of cost function to multi-sequence alignment affords various possibilities. One of the most popular scoring mechanisms for multi-sequence alignment is the *sum-of-pairs* (SP), which refers to the following calculation such that, in the case of multi-sequence alignment for *N* sequences, the multiple alignment score is the summation of the scores of all $N^{x}(N-1)/2$ ordinary pairwise alignments of each pair of input sequences

(seq_i's) at the original candidate multi-sequence alignment [21, 62, 84, 97]. The sum-of-pairs score of the multiple sequence alignment A is defined as Equation 6.9:

$$scr_{SP}(A) = \sum_{1<i,j<n} alignment\left(seq_i, seq_j\right) \tag{6.9}$$

In addition to the standard definition of sum-of-pairs given in [98], an arbitrary weight function *w* can be assigned to each pairwise alignment score *score*(seq_i,seq_j). Hence this form of sum-of-pairs is called *weighted sum-of-pairs* [99]. In the context of this analysis, it is aimed to compare the performance of confidence enhanced costing function with sum-of-pairs in terms of clustering quality metrics, i.e. *inter-* and *intra-cluster distance* and *silhouette measure*.

Multi-Sequence Alignment with sum-of-pairs cost function is performed for Environmental Permit Application use case at cutting level *level=3* for all 25 alignment runs. As shown in Figure 6.67, there is a major parallelism about cluster content with confidence enhanced SA mode such that, sum-of-pairs cost function prefers to assign process alternatives *wabo4, wabo2* and partially *wabo1* to singleton clusters. Additionally, {*wabo1, wabo3, wabo5*}, {*wabo1, wabo5*} and {*wabo2, wabo3*} constitute other frequent cluster contents. According to Figure 6.68, four additional process family tree topologies (i.e. process trees PT24-PT27) are instantiated with lower frequencies. Unlike to confidence enhanced SA mode, sum-of-pairs prefers to detach process alternative *wabo2* from the cluster {*wabo2, wabo3*} as shown at process tree PT10 and to construct a singleton cluster for *wabo2* as process tree PT1. This content shift lowers the frequency of process tree topology PT10 and initiates the other cluster {*wabo1, wabo5*} to merge with the process alternative *wabo3*. Figure 6.69 depicts the most frequent process family tree instance for sum-of-pairs mode.

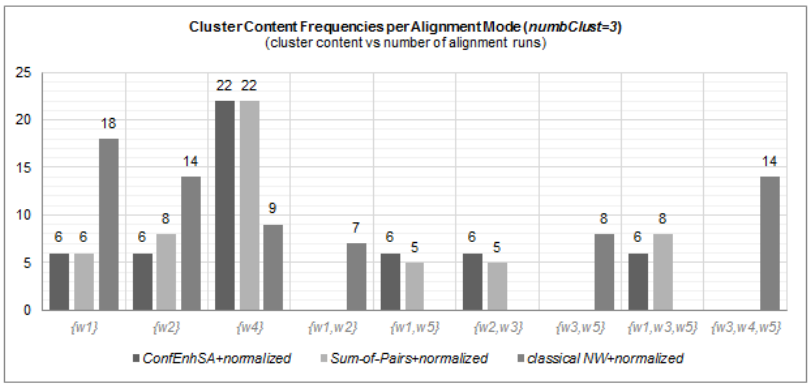


Figure 6.67. Cluster Content Frequencies per Alignment Mode for Environmental Permit Application Use Case.

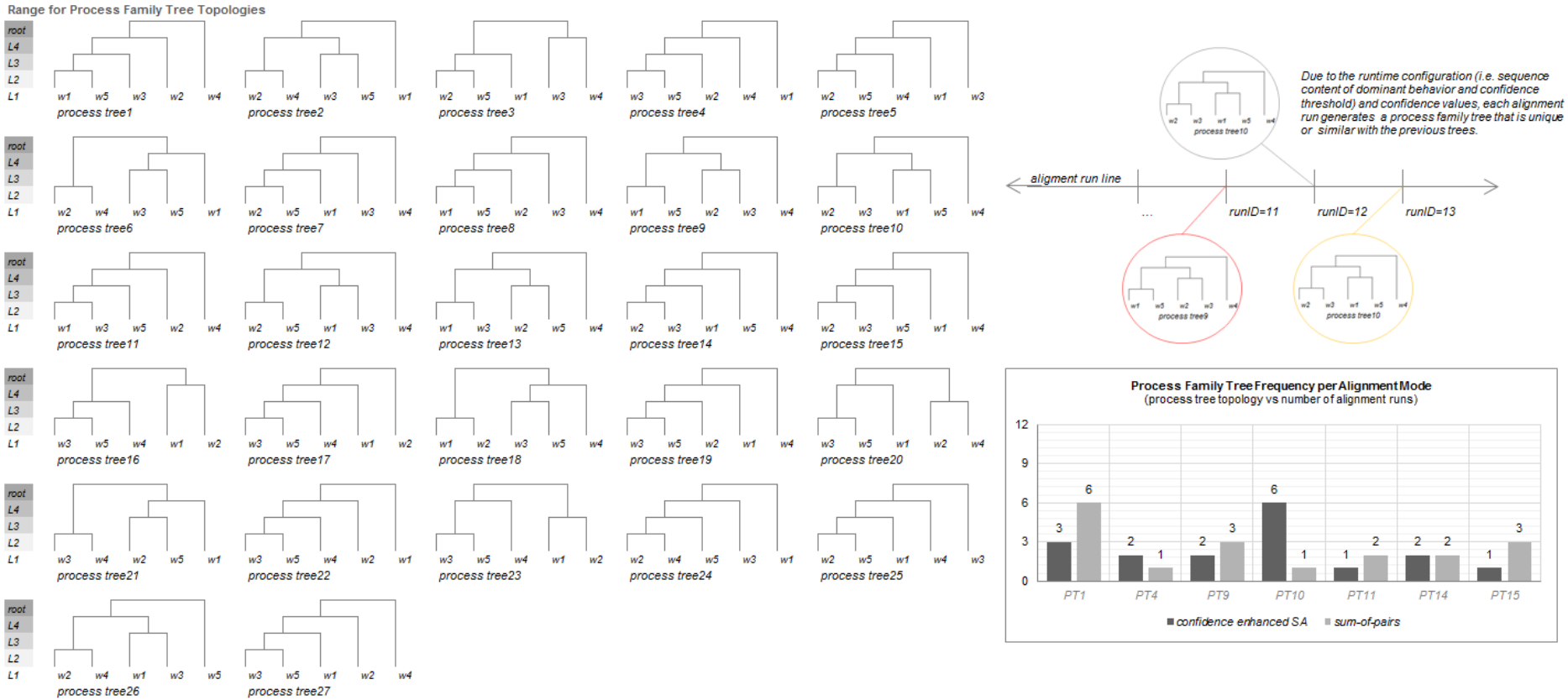


Figure 6.68. Range for Process Family Tree Topologies and Process Family Tree Frequency Obtained by Sum-of-Pairs Mode for Environmental Permit Application Use Case. According to sum-of-pairs, relatively strong relation between the process alternatives *wabo2* and *wabo3* is canceled and this action makes *wabo3* to combine with the prior cluster {*wabo1*, *wabo5*}.

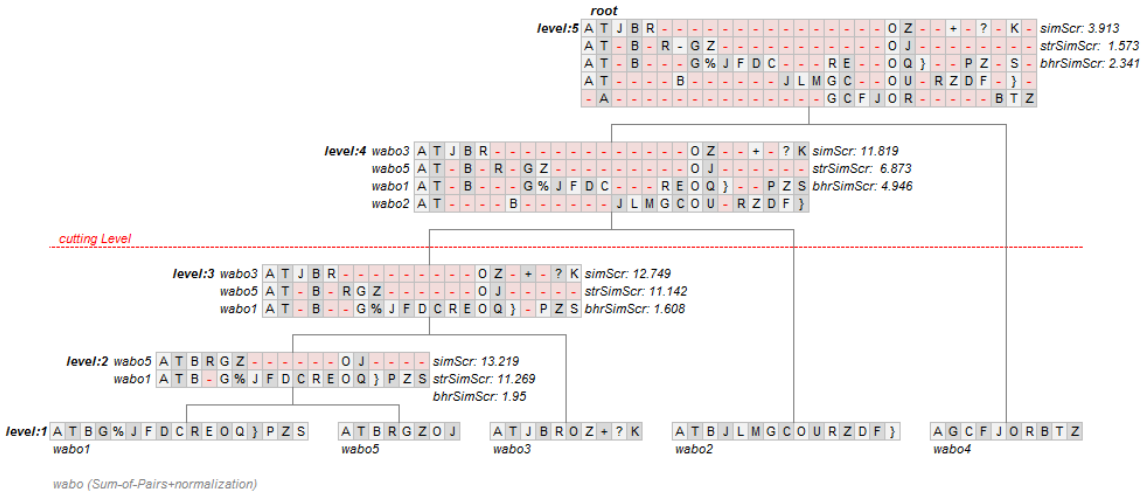


Figure 6.69. Process Family Tree Instance for Multi-Sequence Alignment with Sum-of-Pairs Mode at Environmental Permit Application Use Case. At the cutting level (*level=3*), the clusters {*wabo2, wabo3*}, {*wabo1, wabo5*} *and* {*wabo4*} are instantiated. The underlying process tree topology is shown as *PT1* in Figure 6.68.

Alternatively, the cluster contents at various predefined cutting levels (i.e. *level=3* and *level=4*) and process family tree topologies with respect to sum-of-pairs mode are also evaluated for Period-End Closing use case. As shown at the histogram given in Figure 6.70, due to the industry level business requirements, it enforces the instances of {*client2, client4*} and {*client1, client3, client5*} process clustering according to sum-of-pairs mode. This implies a consensus with prior alignment modes. But according to the cluster contents for *numbCluster=3* setting as shown in Figure 6.71, sum-of-pairs mode deviates from confidence enhanced SA mode by shifting towards the cluster contents {*client2, client4*}, *client5* and {*client1, client3*} instead of *client2*, *client4* and {*client1, client3, client5*} process clusters. This tendency highlights the rationale, according to sum-of-pairs mode, the cohesion between the process alternatives *client2* and *client4* is relatively stronger than the relation between *client5* and the prior cluster {*client1, client3*}. Hence sum-of-pairs mode prefers to assign process alternative *client5* to a singleton cluster rather than *client4* as occurred for confidence enhanced SA mode. Actually, this loose coupling among the process alternative *client5* and the cluster {*client1, client3*} is also emphasized by Multi-Reference Pairwise Alignment technique.

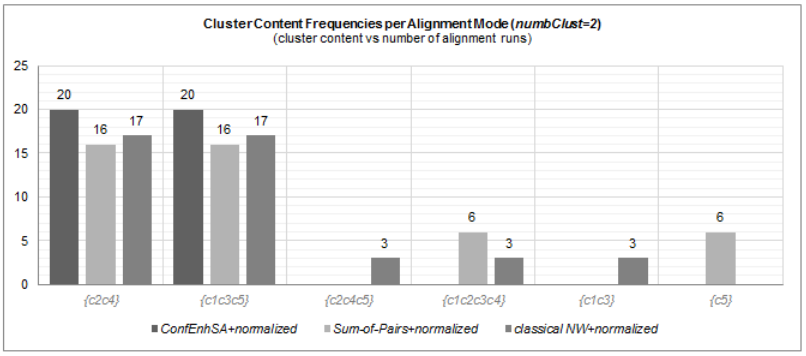


Figure 6.70. Cluster Content Frequencies per Alignment Mode for Period-End Closing Use Case (*numbCluster=2*).

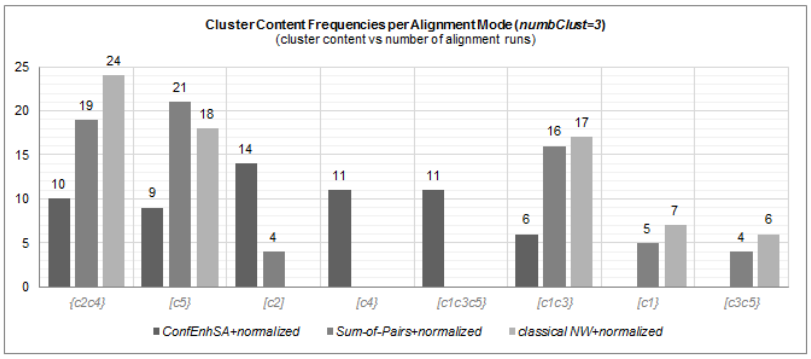


Figure 6.71. Cluster Content Frequencies per Alignment Mode for Period-End Closing Use Case (*numbCluster=3*).

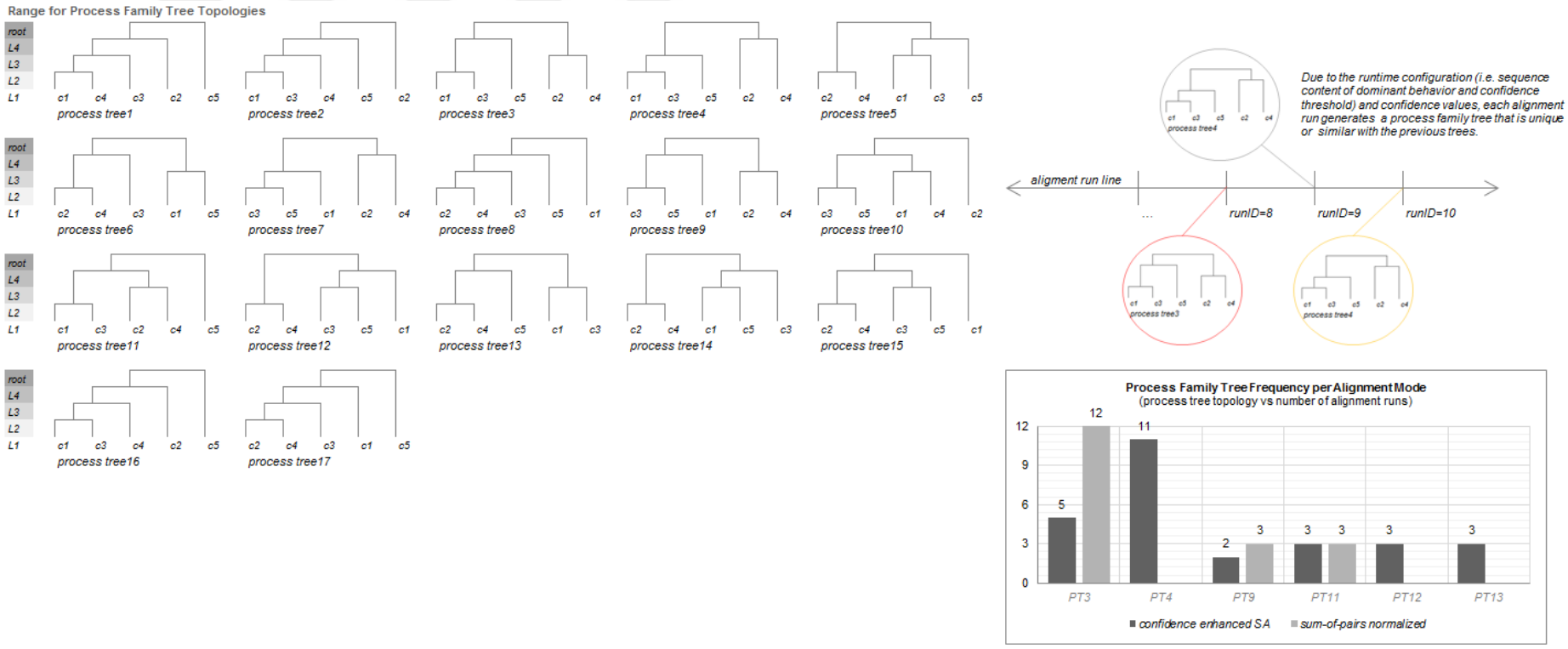


Figure 6.72. Range for Process Family Tree Topologies and Process Family Tree Frequency Obtained by Sum-of-Pairs Mode for Period End Closing Use Case.

As shown in Figure 6.72, the restriction at industry-level business requirements and SAP/CO component differentiation delimit the generation of new process family tree topologies in a such way that, only two new process trees (i.e. process trees PT16 and PT17) with lower frequencies are generated by sum-of-pairs mode. The shrinkage at the frequency of PT4 and the increase at the frequency of PT3 can be explained by the tendency of sum-of-pairs mode towards the late convergence of *client5* to the coherent cluster {*client1, client3*} and sum-of-pairs mode tends to construct alternative process family tree topologies including the singleton cluster of *client5* such as process trees PT1, PT11, PT16 and PT17. While there is a clear difference between confidence enhanced SA and sum-of-pairs modes about the cluster contents at especially cutting level *level=3*, there happens a consensus between these alignment modes about the instance of the most frequent process family tree given in Figures 6.61.

In addition to visualization based on cluster content and process family tree topologies, two distance metrics are adapted to process similarity measurement in order to interpret the quality of process clustering. These metrics aim to distinctly measure how similar the process variant to its neighboring candidates that are assigned to the same process cluster, compared to other process clusters.

***Definition* (inter-cluster distance).** Inter-cluster distance is the normalized form of the inter-cluster similarity, which measures the average cosine similarity between the process clusters, C_i and C_j, which are instantiated at the cutting level *l* of the process family tree, and exists at the cluster range set *CN*. *Process cluster vector* may constitute of one or more process variant vectors introduced in Section 5.2.3 such that, c_i is a process cluster vector holding the similarity score among the all candidate process cluster vector c_j's instantiated at the cutting level *l* and exists at the cluster range set *CN*. The *term weight j* of the process cluster vector c_i (c_{ij}) is the similarity score obtained by aligning the corresponding process clusters c_i and c_j. Technically, inter-cluster distance refers to the coupling between the corresponding process clusters as given in Equation 6.10.

$$inter\_cluster_{dist} = \sqrt{0.5 \times \left(1 - \left(\sum_{C_i, C_j \in CN, i<j} cosSim\left(\vec{c_i}, \vec{c_j}\right) \Big/ C\binom{N}{2}\right)\right)} \tag{6.10}$$

***Definition* (intra-cluster distance).** Intra-cluster distance is the normalized form of the intra-cluster similarity, which measures the average cosine similarity between the base-level (*level=1*) process variants, pv_a and pv_b, that are assigned to the same process cluster c_i. The corresponding process cluster should be *non-singleton* type cluster at the cluster range set *CN*. Process variant vector notation is directly applied for the corresponding process variants as introduced in section 5.2.3, and the cosine of the angle between the corresponding process variant vectors denotes the cohesion among the process variants as given in Equation 6.11.

$$intra\_cluster_{dist} = \sqrt{0.5 \times \left(1 - \left(\sum_{C_i \in CN} \sum_{pv_a, pv_b \in C_i} cosSim\left(\vec{pv_a}, \vec{pv_b}\right)\right) \Big/ \sum_{C_i \in CN} C\binom{|C_i|}{2}\right)} \tag{6.11}$$

Figure 6.73 depicts a sample process cluster similarity and distance measurement for an alignment run that instantiates process tree PT10 at Environmental Permit Application use case. In the context of inter-cluster distance, cosine similarity is calculated according to the similarity scores obtained by pairwise alignment among all process clusters, i.e. *cluster0, cluster1* and *cluster2*. At intra-cluster distance measurement, the pairwise alignments among the base-level process variants at the non-singleton process clusters (i.e. *cluster0* and *cluster1*) are used as the baseline.

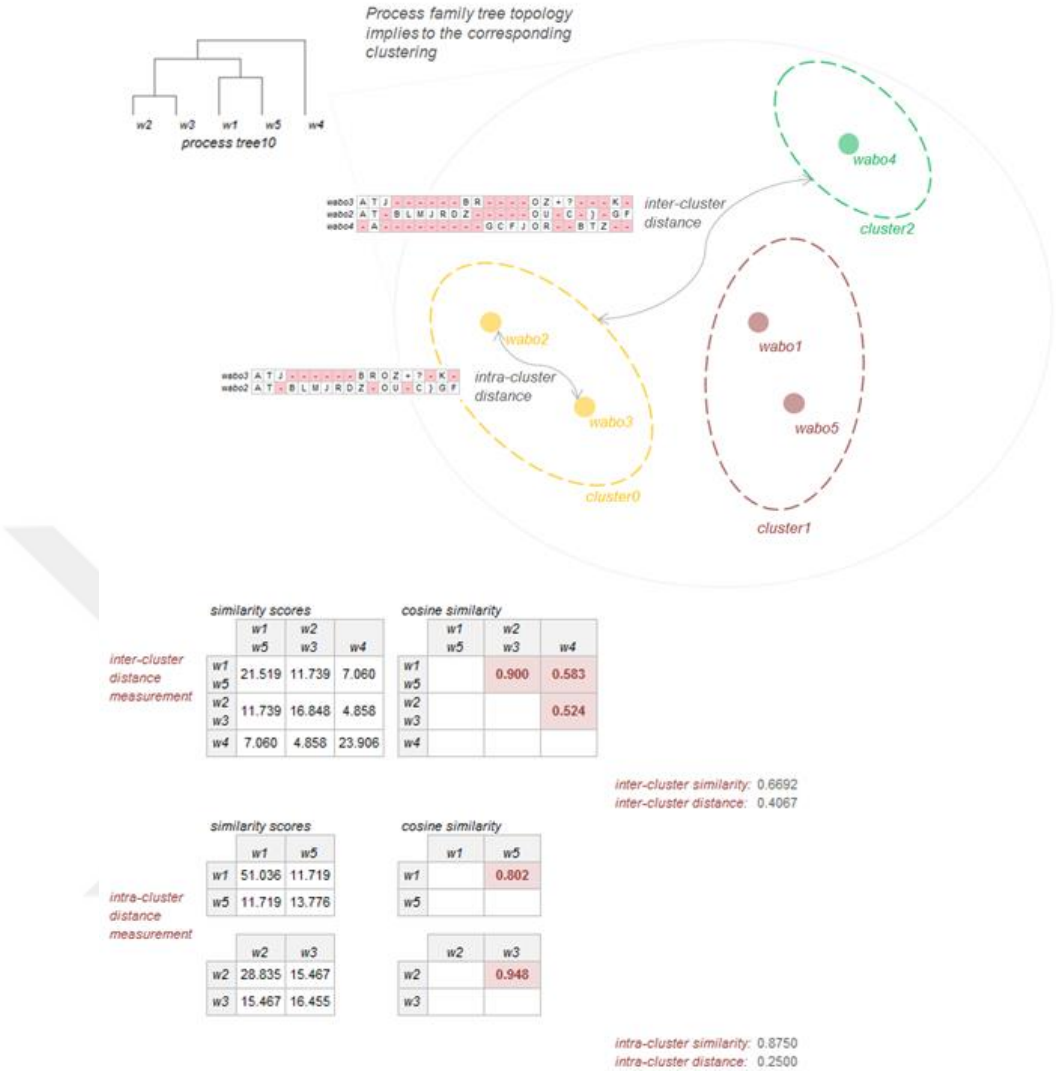


Figure 6.73. A Sample Process Cluster Similarity and Distance Measurement for Environmental Permit Application Use Case.

Intuitively, process clustering with *higher inter-cluster distance* and *lower intra-cluster distance* refers to a good balance at segregating the process alternatives according to their business requirements. As shown in Figure 6.74, while intra-cluster distance for both confidence enhanced SA and sum-of-pairs modes are staggered at [0.685, 0.692] interval, *right-most* distribution of the observations belonged to confidence enhanced SA mode highlights a better discrepancy among uncommon process clusters at Environmental Permit Application use case.

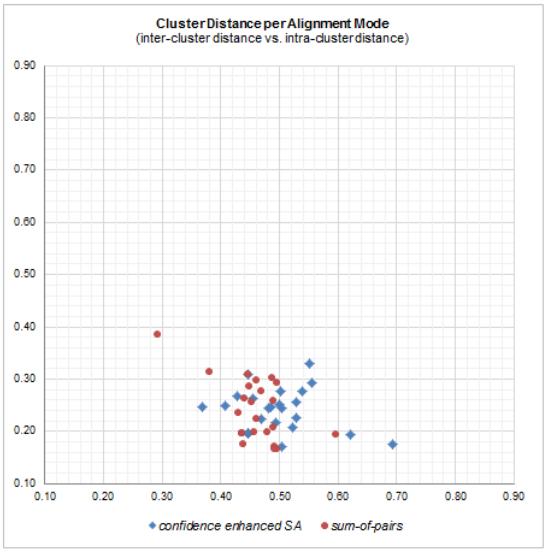


Figure 6.74. Cluster Distance Measurement per Alignment Mode for Environmental Permit Application Use Case (*X-axis:inter-cluster distance value, Y-axis:intra-cluster distance value*). The right-most distribution of confidence enhanced SA mode signals for a better quality at process cluster segregation.

Respectively, the box-plot and whisker charts given in Figures 6.75 and 6.76 emphasize the corresponding mechanism such that, confidence enhanced SA mode shows a relatively normal distribution-like behavior with a higher median value than sum-of-pairs mode for inter-cluster distance (0.699 versus 0.698) as shown in Figure 6.75. On the other hand, the distributions for intra-cluster distance show apparently similar characteristics (e.g. median value and skewness).

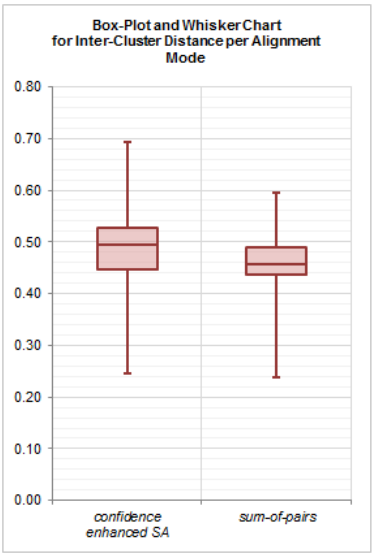


Figure 6.75. Box-Plot and Whisker Chart for Inter-Cluster Distance per Alignment Mode for Environmental Permit Application Use Case.

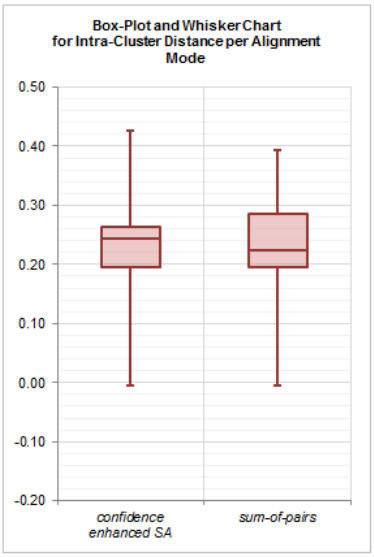


Figure 6.76. Box-Plot and Whisker Chart for Intra-Cluster Distance per Alignment Mode for Environmental Permit Application Use Case.

In addition to visualization of the distance metrics, dependent t-test is applied for interpreting whether there is a significant distinction for the distribution of inter-cluster distance according alignment modes. According to the t-value (3.263 versus $t_{0.05,24}$), the null hypothesis, H_0, which states that there is no clear distinction between the inter-cluster distance measurements per alignment mode, is rejected and the p-value ($p < 0.05$) strengthens this outcome. Positive t-value implies that; process clustering with confidence enhanced SA mode segregates the process families into quite distinct groups within low coupling than sum-of-pairs mode. The result of t-test (α=0.05 and CI=95%) is given in Table 6.21.

Table 6.21. Dependent t-test for Inter-Cluster Distance Measurement for Environmental Permit Application Use Case.

Dependent t-Test Results for Inter-Cluster Distance per Alignment Mode

	conf Enh SA	*sum-of-pairs*
Mean	0,49464	0,45761
Variance	0,00460	0,00269
Observations	25	25
Pearson Correlation	0,57804	
Hypothesized Mean Diff.	0,00000	
DF	24	
t Stat	**3,26266**	
P(T<=t) one-tail	0,00165	
t Critical one-tail	1,71088	
P(T<=t) two-tail	**0,00330**	
t Critical two-tail	**2,06390**	

Similar to Environmental Permit Application use case, similar visualization and statistical analysis are performed for Period End Closing use case. As shown in Figure 6.77, while cohesion due to intra-cluster distance is staggered at [0.695, 0.705] interval, the right-most observations for confidence enhanced SA mode signal for a better performance at segregating the process alternatives at process clusters.

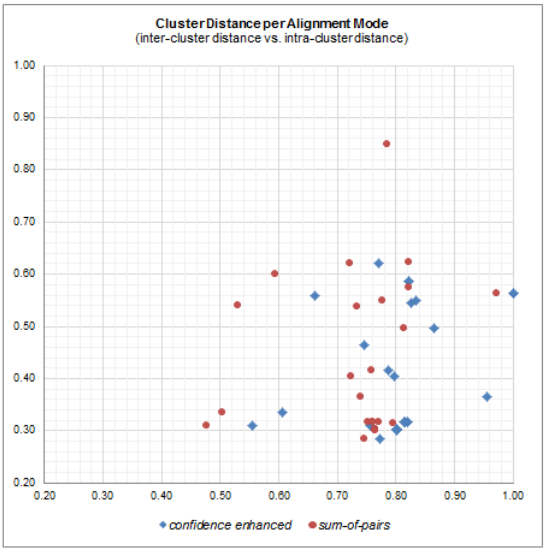


Figure 6.77. Cluster Distance Measurement per Alignment Mode for Period End Closing Use Case (*X-axis:inter-cluster distance value, Y-axis:intra-cluster distance value*).

As an alternative visualization, the underlying mechanism is also interpreted by box-plot and whisker charts given in Figures 6.78 and 6.79. While inter-cluster distance obtained by confidence enhanced SA mode has a higher median value (0.709 versus 0.708) and a wider distribution span (with respect to quartiles 1-3 and outlier values), intra-cluster distance distributions show similar characteristics (median and skewness) except the quartile3 and maximum values.

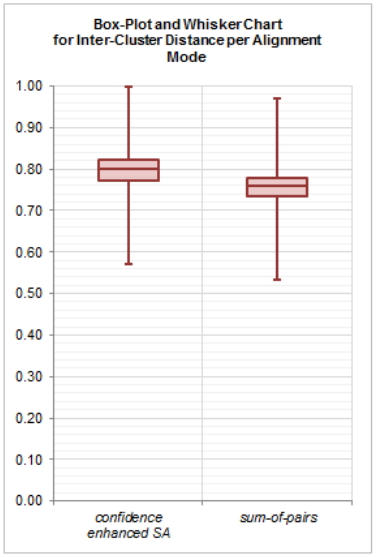


Figure 6.78. Box-Plot and Whisker Chart for Inter-Cluster Distance per Alignment Mode for Period End Closing Use Case.

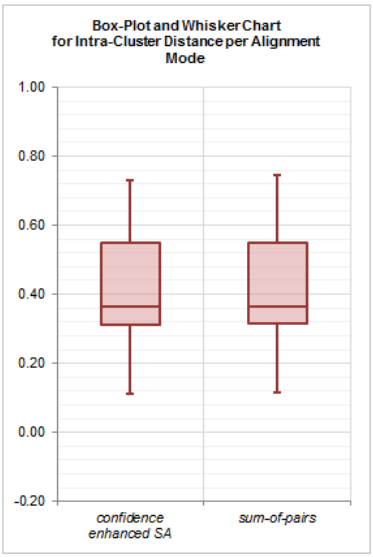


Figure 6.79. Box-Plot and Whisker Chart for Intra-Cluster Distance per Alignment Mode for Period End Closing Use Case.

It is also possible to strengthen the underlying visualization outcomes by statistically analyzing the clustering quality in term of inter-cluster distance metric. According to the t-value (5.114 versus $t_{0.05,24}$) and p-value ($p < 0.05$), the null hypothesis, H_0, which states that there is no clear distinction between inter-cluster distance measurements per alignment mode, is rejected. Positive t-value emphasizes that process clustering performed by confidence enhanced SA mode appropriately segregates the process alternatives into quite distinct process families. The result of t-test (α=0.05 and CI=95%) is given in Table 6.22.

Table 6.22. Dependent t-test for Inter-Cluster Distance Measurement for Period End Closing Use Case.

Dependent t-Test Results for Inter-Cluster Distance per Alignment Mode

	conf Enh SA	*sum-of-pairs*
Mean	0.80190	0.73635
Variance	0.00983	0.01142
Observations	25	25
Pearson Correlation	0.80902	
Hypothesized Mean Diff.	0.00000	
DF	24	
t Stat	**5.11365**	
P(T<=t) one-tail	0.00002	
t Critical one-tail	1.71088	
P(T<=t) two-tail	**0.00003**	
t Critical two-tail	**2.06390**	

While intra- and inter-cluster distance metrics evaluate the quality of process clustering in a dependent manner, these distance metrics are adapted and customized to process similarity measurement problem domain by cosine similarity and distance normalization. Respectively, there are various concepts that combine these inversely correlated metrics and interpret the consistency within the process clusters. In this aspect, the *silhouette measure* argues how similar a process variant is with the neighboring variants at the same process cluster compared to candidate clusters. Higher silhouette values indicate a coherent process clustering with a low coupling among distinct clusters.

***Definition* (silhouette measure).** Let *inter_cl(i)* be the average inter-cluster distance between the clusters at alignment run *i* and *intra_cl(i)* be the average intra-cluster distance among the process alternatives within the same process cluster at alignment run *i*. Silhouette measure is formulated as Equation 6.12:

$$silhouette(i) = \frac{|inter\_cl(i) - intra\_cl(i)|}{max\{inter\_cl(i), intra\_cl(i)\}} \quad (6.12)$$

Respectively for Environmental Permit Application use case, silhouette measure of confidence enhanced SA mode almost dominates all alignment runs with an average value of 0.51 as shown in Figure 6.80 and it has a higher median value within a right-skewed distribution (0.495 versus 0.471) as shown in Figure 6.81. Additionally, it is aimed to statistically analyze the performance of this alignment mode at process clustering by dependent t-test. According to t-value (2.159 versus $t_{0.05,24}$) and p-value ($p < 0.05$), the null hypothesis, H_0, which states that there is no clear distinction between the silhouette measure of the corresponding alignment modes, is rejected. The positive t-value implies that confidence enhanced SA mode generates a more compact process clustering with a good balance between cohesion and coupling. The result of t-test (α=0.05 and CI=95%) is given in Table 6.23.

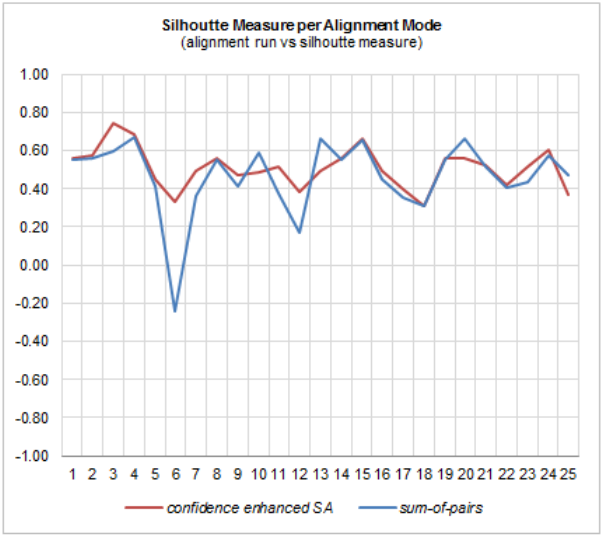


Figure 6.80. Silhouette Measure per Alignment Mode for Environmental Permit Application Use Case (*X-axis:alignment runID, Y-axis:silhouette measure value*).

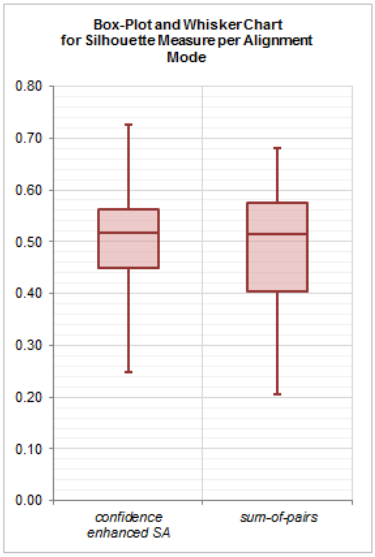


Figure 6.81. Box-Plot and Whisker Chart for Silhouette Measure per Alignment Mode for Environmental Permit Application Use Case.

Table 6.23. Dependent t-test for Silhouette Measurement for Environmental Permit Application Use Case.

Dependent t-Test Results for Silhouette Measure per Alignment Mode

	conf Enh SA	*sum-of-pairs*
Mean	0,50990	0,46531
Variance	0,01126	0,03737
Observations	25	25
Pearson Correlation	0,72069	
Hypothesized Mean Diff.	0,00000	
DF	24	
t Stat	**2,15868**	
P(T<=t) one-tail	0,02042	
t Critical one-tail	1,71088	
P(T<=t) two-tail	**0,04157**	
t Critical two-tail	**2,06390**	

Alternatively, the silhouette measures of the underlying alignment modes are quite analogous at Period End Closing use case except the alignment run 5 as shown in Figures 6.82 and 6.83. While median values for confidence enhanced SA and sum-of-pairs modes are 0.492 and 0.451 respectively, confidence enhanced SA mode has a normal distribution. Respectively, the positive t-value (3.122) obtained at dependent t-test highlights the rationale such that, process clustering by confidence enhanced SA mode tends to assign the neighboring process alternatives sharing common business requirements or rules into the same process families and it segregates distinct process alternatives in a better and appropriate way. The result of t-test (α=0.05 and CI=95%) is given in Table 6.24.

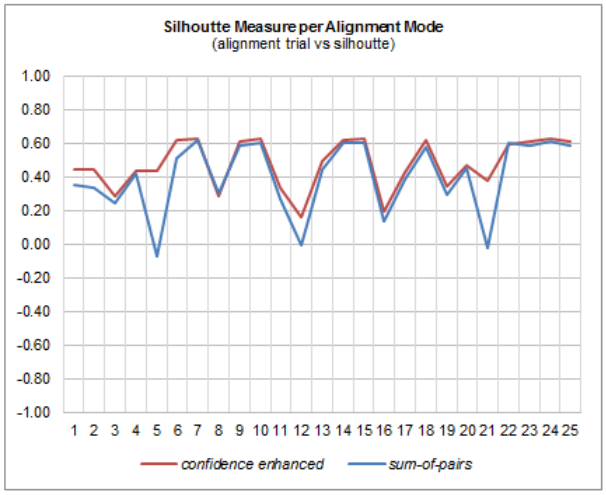


Figure 6.82. Silhouette Measure per Alignment Mode for Period End Closing Use Case (*X-axis:alignment runID, Y-axis:silhouette measure value*).

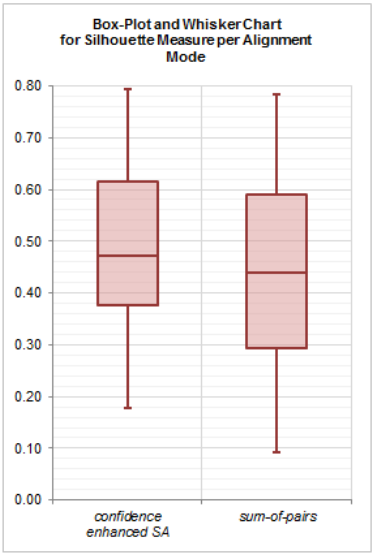


Figure 6.83. Box-Plot and Whisker Chart for Silhouette Measure per Alignment Mode for Period End Closing Use Case.

Table 6.24. Dependent t-test for Silhouette Measurement for Period End Closing Use Case.

Dependent t-Test Results for Silhouette Measure per Alignment Mode

	conf Enh SA	*sum-of-pairs*
Mean	0.47642	0.40018
Variance	0.02207	0.04663
Observations	25	25
Pearson Correlation	0.83845	
Hypothesized Mean Diff.	0.00000	
DF	24	
t Stat	**3.12249**	
P(T<=t) one-tail	0.00232	
t Critical one-tail	1.71088	
P(T<=t) two-tail	**0.00463**	
t Critical two-tail	**2.06390**	

Respectively, sum-of-pairs cost function may turn into an impractical way to handle large sets of process alternatives and multi-sequence alignment with this scoring method turns into NP-complete for longer sequences. Indeed, total process time at confidence enhanced SA mode is approximately 50% shorter than total process time obtained by sum-of-pair mode as shown in Figure 6.84.

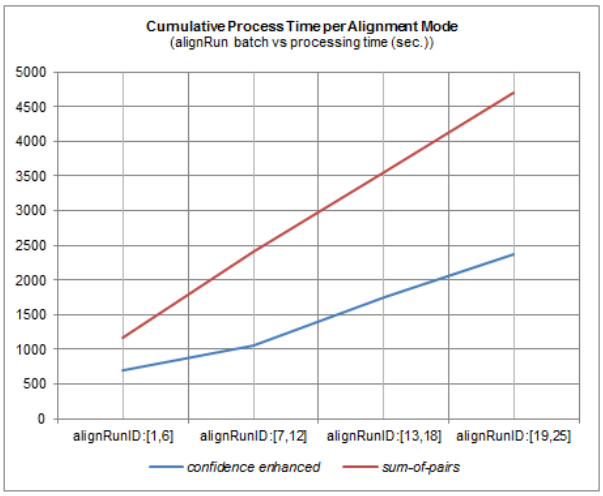


Figure 6.84. Cumulative Processing Time per Alignment Mode for Period End Closing Use Case (*X-axis:alignment run batch, Y-axis:cumulative processing time*).

Consequently, the clustering results for Multi-Reference Pairwise Alignment, Multi-Sequence Alignment with classical NW, sum-of-pairs and confidence enhanced SA modes, prior sum-of-pairs adaptation introduced in [21, 62, 84] and two prior results in the literature [27, 28] that handle Environmental Permit Application use case are cross-validated in Table 6.25. Multi-Reference Pairwise Alignment is consistent with Multi-Sequence Alignment at detecting the *outlier-like* process alternatives, i.e. especially *wabo4* and *wabo1*. Additionally, sum-of-pairs mode tends to generate cross outcomes that harmonize the tendency of both Multi-Reference Pairwise Alignment and Multi-Sequence Alignment with confidence enhanced SA mode. Although, there is not an exact correlation between the proposed approaches and prior studies [27, 28], there happens a consensus on highlighting the commonalities between process alternatives *wabo3* and *wabo5.* Moreover, the singleton clustering results of Multi-Reference Pairwise Alignment and Multi-Sequence Alignment approaches are in parallel with the prior studies.

Table 6.25. Cluster Instances for Multi-Reference Pairwise Sequence Alignment, Multi-Sequence Alignment, Prior Sum-of-Pairs Adaptations given in [21, 62, 84] and Prior Studies given in [27, 28] at Environmental Permit Application Use Case. Respectively, the first clustering content lines refer to relatively stronger results.

Alignment Approach or Clustering Method	Cluster Content
Multi-Reference Pairwise Alignment	{*wabo2*,*wabo3*,*wabo5*}, {*wabo1*}, {*wabo4*}
Multi-Sequence Alignment (*confidence enhanced SA mode*)	{*wabo1*,*wabo5*}, {*wabo2*,*wabo3*}, {*wabo4*} v {*wabo1*,*wabo3*,*wabo5*}, {*wabo2*}, {*wabo4*}
Multi-Sequence Alignment (*sum-of-pairs mode* adapted from [21, 62, 84]	{*wabo1*,*wabo3*,*wabo5*}, {*wabo2*}, {*wabo4*} v {*wabo1*,*wabo5*}, {*wabo2*,*wabo3*}, {*wabo4*} v {*wabo2*,*wabo3*,*wabo5*}, {*wabo1*}, {*wabo4*}
Multi-Sequence Alignment (*classical NW mode*)	{*wabo3*,*wabo4*,*wabo5*}, {*wabo1*}, {*wabo2*} v {*wabo3*,*wabo5*}, {*wabo1*,*wabo2*}, {*wabo4*}
Clustering Content given in [27]	{*wabo1* ,*wabo3* ,*wabo4*},{*wabo2* ,*wabo5*}
Clustering Content given in [28]	{*wabo1*},{*wabo2* ,*wabo4*},{ *wabo3* ,*wabo5*}

Similarly, the clustering results for Multi-Reference Pairwise Alignment, Multi-Sequence Alignment with classical NW, sum-of-pairs and confidence enhanced SA modes at Period End Closing use case are consistent as shown in Table 6.26.

Table 6.26. Cluster Instances for Multi-Reference Pairwise Sequence Alignment and Multi-Sequence Alignment at Period End Closing Use Case.

Alignment Approach or Clustering Method	Cluster Content
Multi-Sequence Alignment (*confidence enhanced SA mode*)	*{client1 ,client3 ,client5},{client2 ,client4}*
Multi-Sequence Alignment (*confidence enhanced SA mode*)	*{client1,client3,client5},{client2,client4}*
Multi-Sequence Alignment (*sum-of-pairs mode* adapted from [21, 62, 84]	*{client1,client3,client5},{client2,client4}*
Multi-Sequence Alignment (*classical NW mode*)	*{client1,client3,client5},{client2,client4}*

Consequently, the fundamental motivation of confidence enhanced SA mode is to eliminate the edit operations that contradict with the underlying business context: while the substitution of contrasting activities and inDel (insertion/deletion) operations of activities with little compatibility for the corresponding business rules should be avoided by dynamically determined penalty scores, the tasks with complementary business circumstance should be encouraged to be substituted or inserted at practical costs.

Respectively, lasagna-like process variants with sparsely filled confidence table tend to be more conservative towards replacement rather than inDel operation that violates the business circumstances. This results in higher structural similarity and inhibits the alignment length at moderate lower levels. On the contrary, spaghetti-like process variants with relatively full confidence tables are feasible for inDel operations. Alternatively according to silhouette measure, which indicates the balance between the intra- and inter-cluster distance, confidence enhanced SA mode instantiates more compact process clustering with maximum cohesion and minimum coupling rather than sum-of-pairs (SP).

6.5. Process Configuration Analysis

Process configuration phase aims to explore the common patterns of activity invocations at process alternatives that are assigned into the same process family at sequence alignment phase. The common patterns are characterized by two feature sets: *identical pairs (IP)* and *maximal identical pairs (maxIP)*.

6.5.1. Alignment Matrix Visualization and Identical Pair Derivation

Before deriving the identical pair feature sets for Environmental Permit Application use case, the alignments at the process families are distinctly visualized by *alignment matrix* on alignment run basis. These matrices decompose the overlapping regions and deviations, i.e. exceptional behaviors that are captured in the regions sparsely filled with the gap symbol (-), and facilitates the interpretation of the conserved regions. Basically, each row indicates the aligned sequences belonging to source and target individual per alignment run and *average coverage* measures the ratio of identical pair span to the total alignment length. Each column holds the activity labels or the gap symbol (-) assigned by inDel operation.

Tables 6.27 and 6.28 show the process families {*wabo2, wabo3*} and {*wabo1, wabo5*} instantiated at the process family tree given in Figure 6.50. As also shown in Figure 6.85, there is a strong correlation between average coverage and structural similarity such that, if the matching operations, which represent substitutive activities encouraging to be replaced according to the business context, dominate the underlying alignment run, the structural similarity tends to increase. Similarly, these matching operations pinpoint potential commonly-used process constructs among the process alternatives.

Table 6.27. Alignment Matrix for {*wabo2, wabo3*} Process Family at Environmental Permit Application Use Case. *Average coverage* refers to total span of IPs at the corresponding alignment run. While gap symbol (–) is highlighted in red color, identical pairs are grey-shaded.

runID	process alternative	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	average coverage
11	wabo3	A	T	J	B	R	-	-	-	O	Z	-	-	+	?	-	K			0.250
	wabo2	A	T	-	B	C	L	M	J	O	U	R	D	Z	G	}	F			
12	wabo3	A	T	J	-	-	-	-	-	-	B	R	O	Z	+	-	?	K		0.176
	wabo2	A	T	-	B	L	M	J	R	D	Z	-	O	U	G	C	F	}		
13	wabo3	A	T	J	-	-	-	-	-	-	B	R	O	Z	+	?	-	K	-	0.167
	wabo2	A	T	-	B	L	M	J	R	D	Z	-	O	U	-	C	}	G	F	
20	wabo3	A	T	J	-	-	-	-	-	-	B	R	O	Z	+	?	-	K	-	0.167
	wabo2	A	T	-	B	L	M	J	R	D	Z	-	O	U	-	C	}	G	F	
22	wabo3	A	T	J	B	R	-	-	-	O	Z	-	-	+	?	-	K			0.250
	wabo2	A	T	-	B	C	L	M	J	O	U	R	Z	D	G	}	F			
25	wabo3	A	T	J	-	B	R	-	-	-	O	Z	-	-	+	?	K	-	-	0.158
	wabo2	A	T	-	B	-	-	L	M	J	O	U	R	Z	D	}	C	G	F	

Table 6.28. Alignment Matrix for *{wabo1, wabo5}* Process Family at Environmental Permit Application Use Case.

runID	process alternative	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	average coverage
11	wabo5	A	T	B	R	G	-	Z	-	-	-	-	-	O	J	-	-	-	-	0.278
	wabo1	A	T	B	-	G	%	J	R	F	D	E	C	O	Q	}	P	Z	S	
12	wabo5	A	T	B	R	G	Z	-	-	-	-	-	-	O	J	-	-	-	-	0.278
	wabo1	A	T	B	-	G	%	F	C	J	R	D	E	O	Q	}	P	Z	S	
13	wabo5	A	T	B	R	G	Z	-	-	-	-	-	-	O	J	-	-	-	-	0.278
	wabo1	A	T	B	-	G	%	F	J	D	C	R	E	O	Q	}	P	Z	S	
20	wabo5	A	T	B	R	G	-	Z	-	-	-	-	-	O	J	-	-	-	-	0.278
	wabo1	A	T	B	-	G	%	J	D	R	F	E	C	O	Q	}	Z	P	S	
22	wabo5	A	T	B	R	G	Z	-	-	-	-	-	-	O	J	-	-	-	-	0.278
	wabo1	A	T	B	-	G	%	F	C	J	R	D	E	O	Q	}	P	Z	S	
25	wabo5	A	T	B	R	G	Z	-	-	-	-	-	-	O	J	-	-	-	-	0.278
	wabo1	A	T	B	-	G	%	F	C	J	R	D	E	O	Q	P	}	Z	S	

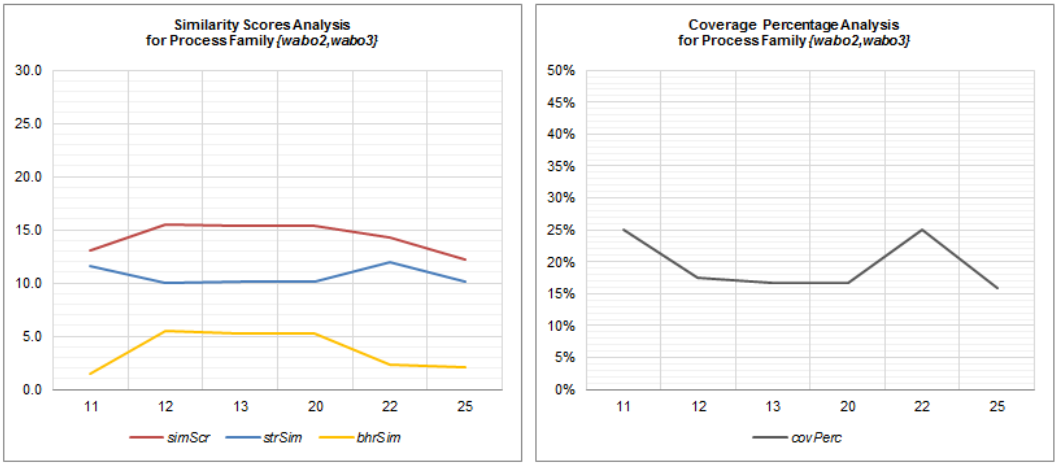


Figure 6.85. Similarity Scores and Coverage Percentage Correlation Analysis for {*wabo2, wabo3*} Process Family at Environmental Permit Application Use Case (*X-axis:alignment runID, Y-axis:similarity score and coverage percentage*).

As the following step, the conserved regions, which are shared by the process alternatives in the same process family, are identified by the feature sets, i.e. identical pairs (IP) and maximal identical pairs (maxIP), for Environmental Permit Application use case. These feature sets are also characterized by various attributes: *order* is the length of the underlying identical pairs, and *frequency* is the repetition of the identical pairs over the selected alignment runs. *Coverage* is the ratio between the order and total alignment length.

Table 6.29 lists all the identical pairs derived at the process families given in Figure 6.50. As the rule of thumb, maxIP should never be subsumed as a substring of any other IPs at alignment runs and order of the underlying IP should exceed 1-unit limit. Hence all IPs except {AT} and {ATB} are eliminated and these two IPs are labeled as maxIP.

Table 6.29. List of Derived Identical Pairs (IP) for Environmental Permit Application Use case.

Process Family	*Identical Pairs (IP)*	*IP in CoSeLoG*	*IP Describtion*	*Order*	*Frequency*	*Coverage*	*maxIP*
wabo3 wabo2	{AT}	{START,770}	*{Start,Establish decision phase of the verdict of court}*	2	1.000	0.117	*yes*
	{B}	{540}	*{Objection to disposal submitted}*	1	0.333	0.021	*no*
	{O}	{730}	*{Contested disposal affected}*	1	1.000	0.058	*no*
wabo5 wabo1	{ATB}	{START,770,540}	*{Start,Establish decision phase of the verdict of court,Objection to disposal*	3	1.000	0.167	*yes*
	{G}	{590}	*{Received request for preliminary verdict}*	1	1.000	0.056	*no*
	{O}	{730}	*{Contested disposal affected}*	1	1.000	0.056	*no*

Likewise in Environmental Permit Application use case, the alignment matrices are also visualized for Period-End Closing use case to facilitate the interpretation of commonalities and deviations among the process alternatives at the same process clusters. Table 6.30 shows the Multi-Sequence Alignment matrix among the clients that are assigned to manufacturing industry cluster. Especially alignment runs that are dominated by maximal identical pairs (maxIPs) with relatively *higher frequency* and *coverage*, e.g. alignment runs 9, 10, 14, 15, 22, 23, 24 and 25, have relatively higher similarity and structural similarity scores. On the other hand, reduction at the coverage of identical pairs (e.g. alignment runs 18-21) is penalized with negative behavioral similarity scores. This implies that inDel operation of alternative activities typically violates strict business rules of Period-End Closing process. Figure 6.86 depicts this correlation between the structural similarity scores and coverage percentage of all IPs on alignment run basis.

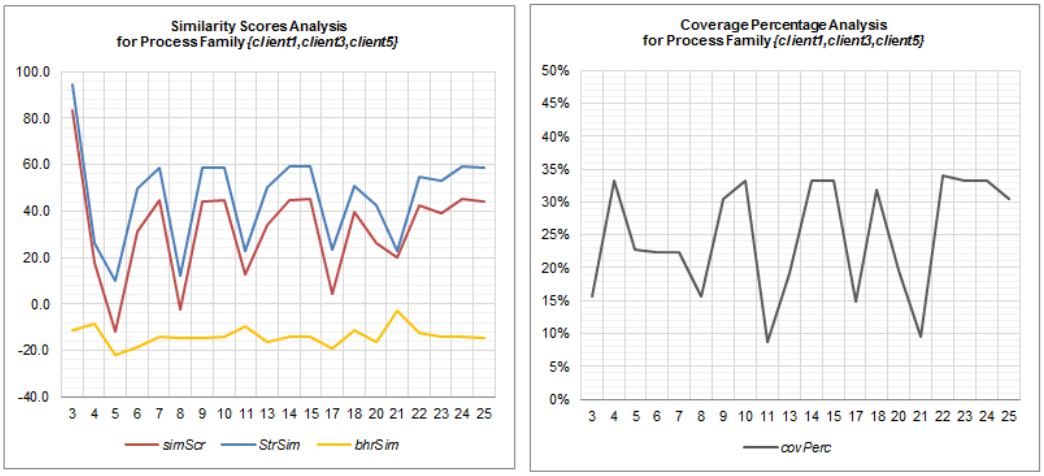


Figure 6.86. Similarity Scores and Coverage Analysis for Process Family {*client1, client3, client5*} at Period-End Closing Use Case (*X-axis:alignment runID, Y-axis:similarity score and coverage percentage*).

As shown in Table 6.31, all relevant alignment runs at service industry cluster are dominated by the maximal identical pairs. Likewise in manufacturing industry cluster, alignment runs with relatively *lower coverage* (e.g. alignment runs 4–6) result in lower similarity scores as shown in Figure 6.87. Negative behavioral similarity scores prove the conservative nature of Period-End Closing process towards inDel operation.

Table 6.30. Sequence Alignment Matrix for Process Family {*client1, client3, client5*} at Period-End Closing Use Case. Especially alignment runs dominated by *maxIPs* (e.g. :[)(+, KCLD and ZXU) and *average coverage*≥0.333 have relatively higher similarity scores stabilized at approximately 60.0 level as shown in Figure 6.86.

runID	process alternative	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	average coverage
3	client5	-	-	-	-	-	-	A	}	%	@	.	Y	}	-	+	(	)	[	-	:	F	-	I	-	D	Z	X	U	\	/	M	-	K	-	L	C	-	-	-	-	-	-	-	-	-										0.156
	client3	-	-	-	-	-	-	A	-	-	@	-	-	-	%	}	O	V	T	?	R	P	Y	}	-	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	(	+										
	client1	A	J	B	]	H	#	-	-	-	@	-	-	-	-	}	O	V	T	-	R	P	?	}	'	Y	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	[	)	(	+										
4	client3	-	-	-	-	-	-	A	-	O	V	T	?	R	P	-	Y	}	Z	X	U	-	-	@	}	-	-	-	M	K	C	L	D	\	I	F	-	:	[	)	(	+	%													0.333
	client1	A	J	B	]	H	#	@	}	O	V	T	-	R	P	?	Y	-	Z	X	U	M	-	'	}	-	-	-	G	K	C	L	D	\	I	F	E	:	[	)	(	+	-													
	client5	-	-	-	-	-	-	-	-	-	-	A	-	-	-	.	Y	}	Z	X	U	\	/	M	-	%	K	@	-	-	C	L	D	-	I	F	-	:	[	)	(	+	}													
5	client3	-	-	-	-	-	-	A	-	O	V	T	R	?	P	Y	}	}	%	@	-	-	-	Z	X	U	M	K	(	+	)	C	L	D	\	I	F	-	:	[	-	-	-	-	-											0.227
	client1	A	J	B	]	H	#	@	}	O	V	T	R	-	-	-	'	}	-	P	?	Y	-	Z	X	U	M	-	-	G	K	C	L	D	\	I	F	E	:	[	)	(	+	-	-											
	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	@	.	Y	}	Z	X	U	-	\	/	M	K	C	L	D	-	I	F	-	:	[	)	(	+	}	%											
6	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	Y	}	Z	X	U	\	/	M	-	K	C	-	-	L	D	-	I	-	-	F	-	:	[	)	(	+						0.224
	client3	-	-	-	-	-	-	A	@	%	}	O	V	T	R	?	P	Y	-	-	-	-	-	}	-	-	Z	X	U	-	-	M	-	K	C	+	(	L	D	\	I	)	[	F	-	:	-	-	-	-						
	client1	A	J	B	]	H	#	-	@	-	}	O	V	T	R	-	P	?	'	-	-	-	-	}	Y	-	Z	X	U	-	-	M	G	K	C	-	-	L	D	\	I	-	-	F	E	:	[	)	(	+						
7	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	Y	}	Z	X	U	\	/	M	-	K	C	-	-	L	D	-	I	-	-	F	-	:	[	)	(	+						0.224
	client3	-	-	-	-	-	-	A	@	%	}	O	V	T	R	?	P	Y	-	-	-	-	-	}	-	-	Z	X	U	-	-	M	-	K	C	+	(	L	D	\	I	)	[	F	-	:	-	-	-	-						
	client1	A	J	B	]	H	#	-	@	-	}	O	V	T	R	-	P	?	'	-	-	-	-	}	Y	-	Z	X	U	-	-	M	G	K	C	-	-	L	D	\	I	-	-	F	E	:	[	)	(	+						
8	client5	-	-	-	-	-	-	A	}	%	@	.	Y	}	-	+	(	)	[	-	:	F	-	I	-	D	Z	X	U	\	/	M	-	K	-	L	C	-	-	-	-	-	-	-	-	-										0.156
	client3	-	-	-	-	-	-	A	-	-	@	-	-	-	%	}	O	V	T	?	R	P	Y	}	-	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	(	+										
	client1	A	J	B	]	H	#	-	-	-	@	-	-	-	-	}	O	V	T	-	R	P	?	}	'	Y	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	[	)	(	+										
9	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	Y	}	Z	X	U	\	/	M	-	K	C	L	D	-	I	F	-	:	[	)	-	(	+									0.304
	client3	-	-	-	-	-	-	A	@	%	}	O	V	T	?	R	P	Y	}	-	-	-	-	-	-	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	-	(	+									
	client1	A	J	B	]	H	#	-	@	-	}	O	V	T	-	R	P	?	}	-	-	-	-	'	Y	-	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	-	)	[	(	+									
10	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	Y	}	Z	X	U	\	/	M	-	K	C	L	D	-	I	F	-	:	[	)	(	+										0.333
	client3	-	-	-	-	-	-	A	%	@	}	O	V	T	?	R	P	Y	}	-	-	-	-	-	-	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	(	+										
	client1	A	J	B	]	H	#	-	-	@	}	O	V	T	-	R	P	?	}	-	-	-	-	'	Y	-	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	[	)	(	+										
11	client5	-	-	-	-	-	-	A	}	%	@	-	.	-	-	-	-	-	-	-	-	X	Z	}	-	U	Y	-	-	-	-	-	\	/	M	-	K	C	L	D	I	F	:	[	)	(	+									0.087
	client3	-	-	-	-	-	-	A	-	%	@	}	O	V	T	R	?	P	Y	}	-	-	Z	X	:	-	-	-	F	-	I	\	D	[	)	L	U	C	-	-	-	K	M	-	-	(	+									
	client1	A	J	B	]	H	#	-	-	-	@	}	O	V	T	R	-	P	?	}	'	Y	Z	X	F	-	-	I	-	E	U	\	D	-	M	G	K	-	L	-	-	C	:	[	)	(	+									
13	client5	-	-	-	-	-	-	A	}	%	-	@	.	-	-	-	-	-	-	-	-	-	-	X	U	\	/	M	Z	Y	}	K	C	L	D	-	-	-	-	-	I	F	-	:	[	)	(	+								0.191
	client3	-	-	-	-	-	-	A	@	%	}	-	O	V	T	R	?	P	Y	}	-	-	Z	X	U	-	-	M	-	-	-	K	C	L	D	\	-	-	-	-	I	F	-	:	[	)	(	+								
	client1	A	J	B	]	H	#	-	@	-	}	-	O	V	T	R	-	P	?	}	'	Y	Z	X	U	M	-	-	-	-	-	-	-	-	D	\	I	L	C	G	K	F	E	:	[	)	(	+								
14	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	Y	}	Z	X	U	\	/	M	-	K	C	L	D	-	I	F	-	:	[	)	(	+										0.333
	client3	-	-	-	-	-	-	A	@	%	}	O	V	T	R	?	P	Y	}	-	-	-	-	-	-	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	(	+										
	client1	A	J	B	]	H	#	-	@	-	}	O	V	T	R	-	P	?	}	-	-	-	-	'	Y	-	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	[	)	(	+										
15	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	Y	}	Z	X	U	\	/	M	-	K	C	L	D	-	I	F	-	:	[	)	(	+										0.333
	client3	-	-	-	-	-	-	A	@	%	}	O	V	T	R	?	P	Y	}	-	-	-	-	-	-	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	(	+										
	client1	A	J	B	]	H	#	-	@	-	}	O	V	T	R	-	P	?	}	-	-	-	-	'	Y	-	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	[	)	(	+										
17	client1	A	}	@	#	H	[	)	(	:	E	O	]	B	J	V	+	-	R	P	?	-	-	T	-	F	-	}	'	Y	Z	X	U	-	-	M	G	K	I	\	D	C	L	-	-	-	-	-	-	-	-	-	-	-	-	0.148
	client3	A	-	-	-	-	-	-	-	-	-	-	-	-	O	V	T	?	R	-	-	}	-	@	%	P	Y	}	-	-	Z	X	U	-	-	M	-	K	-	-	-	C	L	D	-	+	(	-	)	[	-	\	:	F	I	
	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	-	.	Y	}	-	-	Z	X	U	\	/	M	-	K	-	-	-	C	L	D	I	F	:	[	)	(	+	-	-	-	-	
18	client5	-	-	-	-	-	-	A	-	-	}	%	@	.	-	-	-	-	-	-	-	-	-	-	X	U	\	/	M	Z	}	Y	K	C	L	D	-	I	F	-	:	[	)	(	+											0.318
	client3	-	-	-	-	-	-	A	@	%	}	-	-	O	V	T	?	R	P	-	Y	}	-	Z	X	U	M	-	-	-	-	-	K	C	L	D	\	I	F	-	:	[	)	(	+											
	client1	A	J	B	]	H	#	-	@	-	}	-	-	O	V	T	-	R	P	'	?	}	Y	Z	X	U	M	-	-	-	-	G	K	C	L	D	\	I	F	E	:	[	)	(	+											
20	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	Y	}	Z	X	U	\	/	M	K	C	L	-	D	-	-	-	-	I	F	-	:	[	)	-	(	+				0.196
	client3	-	-	-	-	-	-	-	-	A	V	T	O	?	R	P	-	-	-	Y	}	-	}	%	@	-	-	-	Z	X	U	-	-	M	K	C	L	-	D	\	-	-	-	I	F	-	:	[	)	-	(	+				
	client1	A	J	B	]	H	#	@	}	O	V	T	-	-	R	P	?	}	'	Y	Z	-	-	-	-	-	-	-	-	-	-	-	-	-	K	C	L	G	D	\	X	M	U	I	F	E	:	-	)	[	(	+				
21	client5	-	-	-	-	-	-	A	-	-	}	%	@	.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Z	X	U	-	-	\	/	M	}	Y	K	C	I	L	D	-	F	:	-	[	-	)	(	+		0.094
	client3	-	-	-	-	-	-	A	@	%	}	-	-	O	V	T	R	P	?	-	-	Y	}	Z	U	X	M	K	+	C	L	-	-	-	D	\	-	-	-	-	-	-	-	(	-	I	F	:	)	[	-	-	-	-		
	client1	A	J	B	]	H	#	-	@	-	}	-	-	O	V	T	R	P	}	'	?	Y	-	Z	-	-	G	K	-	C	L	X	U	M	D	\	I	-	-	-	-	-	-	-	E	-	F	:	-	[	(	)	-	+		
22	client5	-	-	-	-	-	-	A	}	%	@	.	Y	Z	-	-	-	-	-	-	-	-	-	-	-	}	X	U	\	/	M	-	K	C	L	D	-	I	F	-	:	[	)	(	+											0.341
	client3	-	-	-	-	-	-	A	-	%	@	}	-	-	V	O	-	T	?	R	P	Y	-	}	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	(	+											
	client1	A	J	B	]	H	#	-	-	-	@	}	-	-	-	O	V	T	-	R	P	?	'	}	Y	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	[	)	(	+											
23	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	-	}	Z	-	Y	-	X	U	\	/	M	-	K	C	L	D	-	I	F	-	:	[	)	(	+										0.333
	client3	-	-	-	-	-	-	A	@	%	}	O	V	T	R	?	-	-	-	-	P	Y	}	-	-	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	(	+										
	client1	A	J	B	]	H	#	-	@	-	}	O	V	T	R	-	-	-	-	-	P	?	}	-	'	Y	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	[	)	(	+										
24	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	Y	}	Z	X	U	\	/	M	-	K	C	L	D	-	I	F	-	:	[	)	(	+										0.333
	client3	-	-	-	-	-	-	A	%	@	}	O	V	T	?	R	P	Y	}	-	-	-	-	-	-	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	(	+										
	client1	A	J	B	]	H	#	-	-	@	}	O	V	T	-	R	P	?	}	-	-	-	-	'	Y	-	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	[	)	(	+										
25	client5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	}	%	@	.	Y	}	Z	X	U	\	/	M	-	K	C	L	D	-	I	F	-	:	[	)	-	(	+									0.304
	client3	-	-	-	-	-	-	A	@	%	}	O	V	T	?	R	P	Y	}	-	-	-	-	-	-	-	Z	X	U	-	-	M	-	K	C	L	D	\	I	F	-	:	[	)	-	(	+									
	client1	A	J	B	]	H	#	-	@	-	}	O	V	T	-	R	P	?	}	-	-	-	-	'	Y	-	Z	X	U	-	-	M	G	K	C	L	D	\	I	F	E	:	-	)	[	(	+									

Table 6.31. Sequence Alignment Matrix for Process Family {*client2, client4*} at Period-End Closing Use Case. Diminish at the coverage returns with shrink at total and structural similarity values as shown in Figure 6.87.

runID	process alternative	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	average coverage
3	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
4	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
5	client2	{	Z	O	N	W	S	R	Y	-	V	T	-	-	-	-	Q	P	)	(	+	0.250
	client4	-	-	-	-	-	-	-	-	O	V	T	R	P	Y	{	Z	-	)	(	+	
6	client2	-	-	O	-	V	-	T	-	R	P	Y	{	Z	)	(	+					0.500
	client4	{	Z	O	N	V	W	T	S	R	-	Y	Q	P	)	(	+					
7	client2	{	Z	O	N	V	W	T	S	R	-	Y	Q	P	)	(	+					0.500
	client4	-	-	O	-	V	-	T	-	R	P	Y	{	Z	)	(	+					
8	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
9	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
10	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
11	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
13	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
14	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
15	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
17	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
18	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
20	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				0.471
	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				
21	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				0.471
	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				
22	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
23	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
24	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				
24	client2	{	Z	O	N	V	T	W	S	R	-	-	Y	Q	P	)	(	+				0.471
	client4	-	-	O	-	V	T	-	-	R	P	{	Y	-	Z	)	(	+				

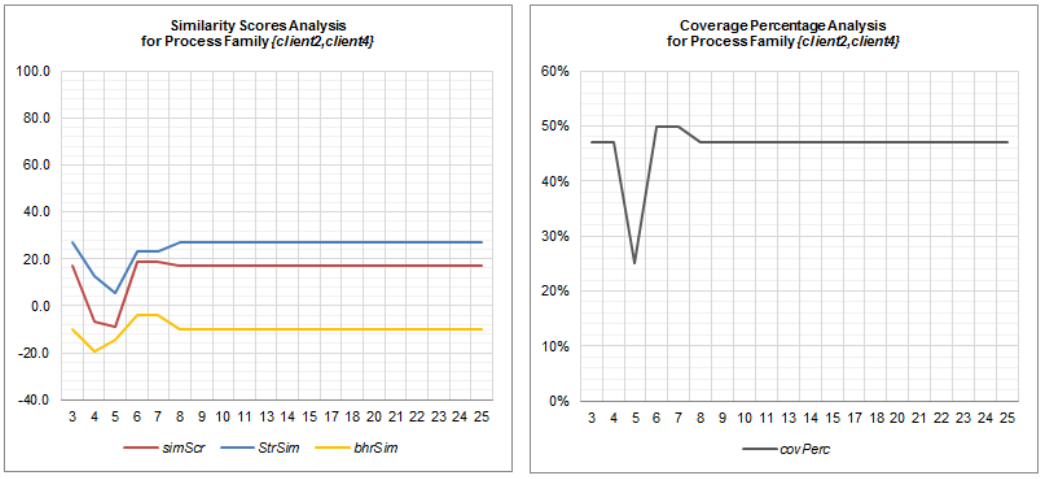


Figure 6.87. Similarity Scores and Coverage Analysis for Process Family {*client2 ,client4*} at Period-End Closing Use Case (*X-axis:alignment runID, Y-axis:similarity score and coverage percentage*).

Table 6.32 summarizes maximal identical pairs (maxIP) feature sets for each process family. Especially maxIPs with relative higher frequency and coverage can be interpreted as a significant

evidence of common behavior and manifestation of this commonality might indicate functional inheritance at the process enactments of organizations in the same process family.

Table 6.32 Maximal Identical Pairs Feature Sets per Process Family for Period-End Closing Use Case.

Process Family	Identical Pairs (IP)	IP Description	Order	Frequency	Coverage	maxIP
client1 client3 client5	{@}	{COGI}	1	0.200	0.004	no
	{ZXU}	{KSV5,KSS2_PRD, KSII_PRD}	3	0.650	0.042	yes
	{M}	{CO88}	1	0.700	0.015	no
	{K}	{CKMLCP_SEL_PRD}	1	0.200	0.004	no
	{IF}	{CKMLCP_RVL_PRD, CKMLCP_PST_PRD}	2	0.600	0.026	yes
	{:])(+}	{FS0N,CKME,MMRV, MMPV,OKP1}	5	0.450	0.048	yes
	{KCLD}	{CKMLCP_SEL_PRD, CKMLCP_DTR_PRD, CKMLCP_SNG_PRD, CKMLCP_MLT_PRD}	4	0.450	0.039	yes
	{(+}	{MMPV,OKP1}	2	0.200	0.009	no
	{XU}	{KSS2_PRD,KSII_PRD}	2	0.200	0.009	no
client2 client4	{O}	{KO8G_INT_MNT}	1	0.950	0.056	no
	{R}	{KON2}	1	0.200	0.012	no
	{VT}	{KSS2_MNT,KSII_MNT}	2	0.900	0.106	yes
	{Y}	{KSU5}	1	0.200	0.012	no
	{)(+}	{MMRV,MMPV,OKP1}	3	1.000	0.176	yes

6.5.2. Configurable Process Modeling

Common regions that are pinpointed by maxIPs can be conceptualized as *abstraction* or *sub-processes* at higher level of process configurations. The deviations or variations among process alternatives are dealt with configurable elements. While applying the corresponding abstraction, two quality dimensions should also be taken into consideration: *generalization* and *simplicity*. Generalization is a desirable feature over observed behavior within a two-sided aspect such that, *underfitting* process model tends to over-generalize the obtained behavior from event logs while *overfitting* ones generate highly-specific outcomes that attempt to explain both distinct deviations and low-frequent patterns [23]. *Simplicity* is another quality dimension which can be enhanced with *Occam's Razor*.

As shown in Figure 6.88, the maxIP {*STRT, 770, 540*} is encapsulated as a sub-process (aka. WABO_INIT) at the primitive form of configurable process model. The corresponding process variants, i.e. *wabo1* and *wabo3*, are obtained by configuring the generic process model elements. As stated in [13], configuration implies the removal of the possibilities and customization of the process according to the process-specific business rules.

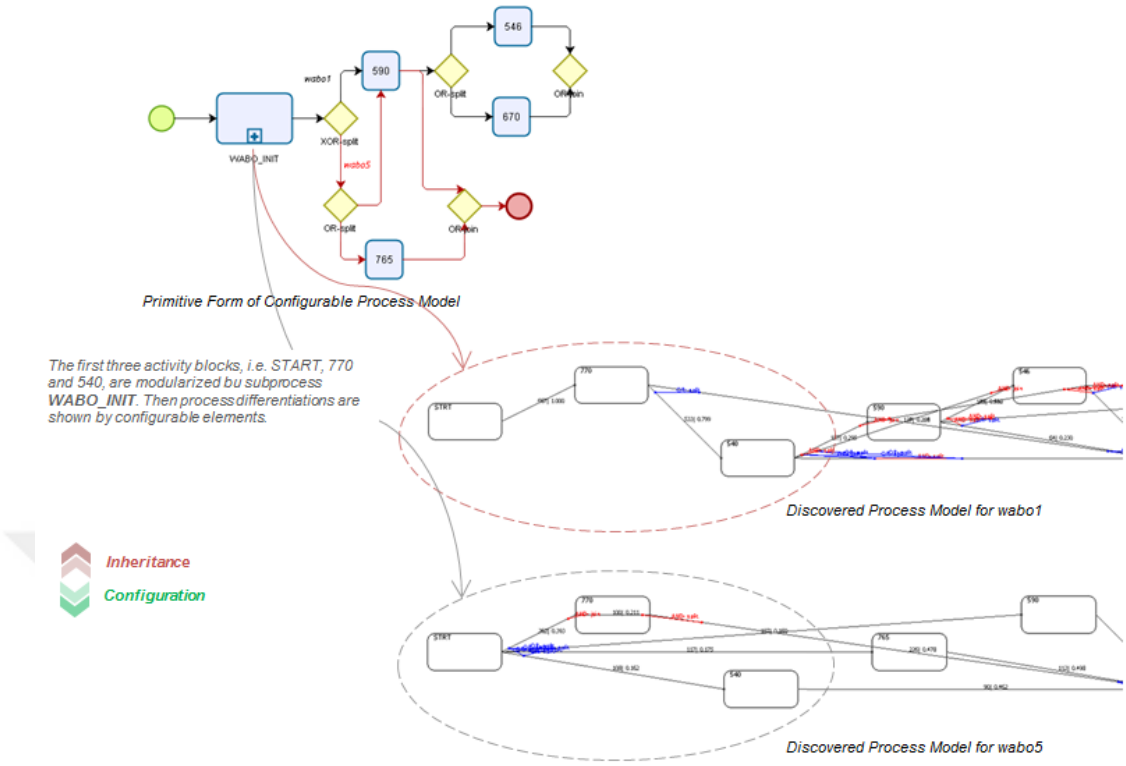


Figure 6.88. Primitive Form of Configurable Process Model for Environmental Permit Application Process in {*wabo1, wabo5*} Process Family.

Alternatively, Figure 6.89 depicts the configurable process model designated for Period-End Closing process devoted to the service industry. While conserved regions, which are pinpointed by maxIP feature set, are encapsulated as sub-processes, divergence among client2 and client4 are handled by configurable elements (e.g. XOR branching). While the maxIP {*KSS2_MNT, KSII_MNT*} is renamed as SRV_MAINTORD_MANG, {*MMRV, MMPV, OKP1*} is encapsulated as SRV_PER_CLS.

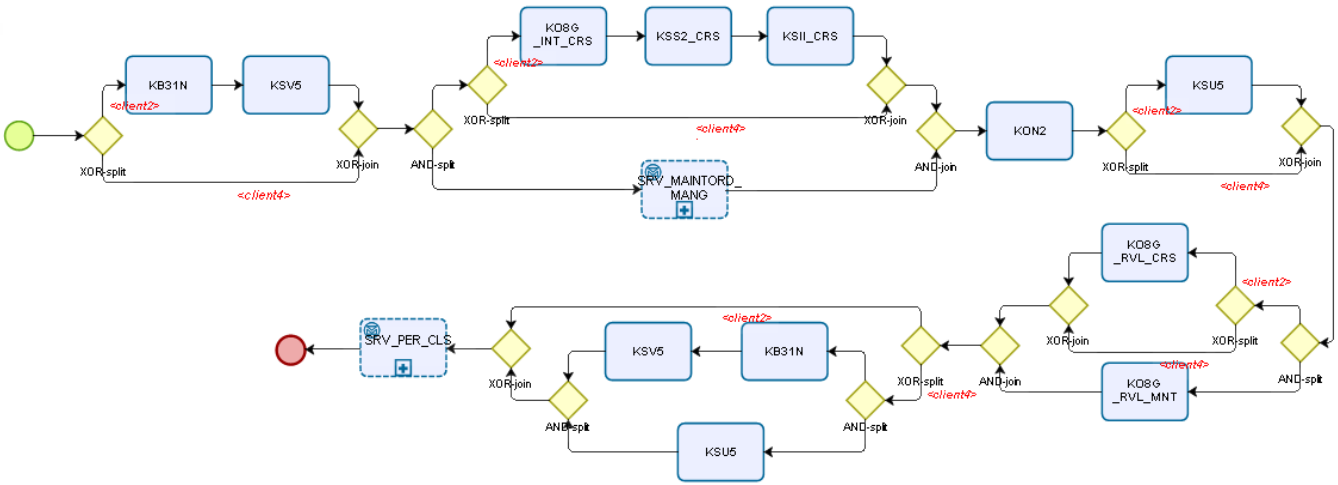


Figure 6.89. Configurable Process Model for Period-End Closing Process in Service Industry.

CHAPTER 7

CONCLUSION

7.1. Summary and Concluding Remarks

Organizations executing the same processes at a shared architecture call for a more systematic treatment to deal with the variability across these organizations. The paradigm shift at the process-aware information systems initiates a new era of process mining, called *cross-organizational process mining*. One of the challenging issues in this era is exploiting the commonalities, which act as the baseline for configurable process models [11, 15, 16]. Current aspects in process similarity measurement are mainly dedicated to *model-model similarity measurement* which implies the adaptations of informational retrieval and graph theory algorithms to semantic, syntactical comparison of the task labels and process structures [19, 21]. Unfortunately, these adaptations are inadequate to interpret the process behaviors in the context of process structures and task labels. Alternatively, *log-model similarity measurement* reflects these confusing process dynamics by instantiating the state space or enumerating all possible process traces [17, 20, 50]. While various equivalence notions can be scaled to this exhaustive enumeration and reflect the moment of choice at process behavior, the atomic true/false response misleads the degree of similarity.

In this aspect, we aim to develop a cross-organizational process mining framework for extracting the similarities among distinct organizations that execute exactly the same business processes. As the following step, these organizations are clustered into *process families* by the adaptations of NW algorithms on the basis of *log-log similarity measurements.* The following results and contributions are handled in the context of corresponding phases.

7.1.1. Dominant Behavior Extraction Results and Contributions

Dominant behavior extraction phase initially derives the exemplary sequence that decodes the dominant behavior, which is the most common sequence of behavior captured at the event logs due to the repetitive occurrence within or across the process instances with high domain significance. This new perspective at process diagnostics simplifies the process similarity measurement by analyzing just this extracted common behavior and thereby bypasses the ultimate requirement for apriori reference process models.

As stated in the research questions given in Section 2.4, encapsulating all of the high-order process behaviors within a single sequence is seemingly inadequate. Therefore, the inter-dependencies among the consecutive activity pairs that share an incorporated business context are traced by *confidence values*. This confidence concept is introduced in [42], then revised as *average confidence* (avgConfFTC) denoted in Equation 5.3.

In order to evaluate the effect of dominant behavior on process discovery performance, two major issues (i.e. dealing with incompleteness and further abstraction) are revisited by the conformance checker supporting two metrics: *completeness* and *soundness* as given in Equations 6.1 and 6.2 respectively. For the corresponding use cases, *high completeness* and *low soundness* at process discovery emphasize a good balance with respect to the quality dimensions. In addition to the balance among completeness and soundness, we introduce a *structural influence factor set* in order to analyze the understandability of the process models: *connectivity, density* and *average transition length* (ATL) as given in Section 6.3.1. Respectively, process candidates with *higher connectivity* and *lower density* tend to generate *spaghetti-like process models* that are hard to interpret by process observers. Therefore this characteristic increases the risk of pruning by confidence/support threshold and results in the loss of process behavior at process discovery. Unlike to spaghetti-like process models, *lasagna-like* process candidates with *lower connectivity* and *higher density* are more robust to the increases at confidence threshold parameter. Moreover, *lower ATL* highlights the mechanism such that, dominant behavior tends to encourage compactness by assigning the activity pairs with stronger succession to consecutive neighboring positions at the sequence. This outcome initiates mostly *straight-line* (*direct successive*) *transitions* at the process discovery as stated in Section 4.1.1.

Another enhancement at dominant behavior extraction phase is the *GA engine adaptation* that aims to find the dominant behavior with the *fittest solution*. Unlike to prior brute-force approach introduced in [42], three drivers are handled to interpret the performance and robustness of GA engine in Section 6.3.2: *schema application*, *crossover probability* and *population size*. Indeed, the rationale hindered by Holland's schema theorem is validated by statistical tests, which implies that process discovery runs with schema requires less iterations to reach to the population convergence according to the difference between the maximal and average fitness scores. On the other hand, the main criticism about schema theory is the assumption that ignores the effect of crossover and mutation framework at the genetic variation. Alternatively, process discovery with lower crossover probability and limited population size tend to behave like myopic local search due to the risk of congesting at local optimal points. On the contrary, opposite GA configuration has a better performance in exploring the search space by the effect of genetic diversity. Then smaller improvements happen when most individuals become quite similar at the following populations.

7.1.2. Sequence Alignment and Process Configuration Results and Contributions

At sequence alignment phase, two adaptations of NW algorithm are employed to measure the degree of similarity between the process alternatives. These adaptations, namely *Pairwise Alignment* and *Multi-Sequence Alignment*, are configured according to three distinct settings: *Single-* and *Multi-Reference Pairwise Alignment*, *Multi-Sequence Alignment*.

At Single-Reference Pairwise Alignment, the intuitive judgments in the form of similarity rankings are collected by the questionnaires and these rankings are converted to the likert-charts as the ground truth. Then various informational retrieval metrics, i.e. *cosine similarity* and *discount cumulative gain*, are adapted to measure the correlation between the proposed approaches and intuitive judgments. As shown in Figures 6.26 and 6.27, Single-Reference Pairwise Alignment is highly correlated with the perceptions of process observers better than the prior approaches given in [66, 67]. It is also validated the quality of similarity measurements with respect to *recall* and *precision* framework emphasized in [91]. Respectively, Single-Reference Pairwise Alignment shows a good balance between recall and precision with a higher mean value (e.g. 0.602 at Travel Management and 0.534 at Loan Application use case) as shown in Figures 6.28 and 6.30. Alternatively, there is a strong positive correlation between the scoring of Single-Reference Pairwise Alignment and the tacit similarity assessment mechanism of more experienced process observers due to solid improvements at the average precision value (i.e. 23.75% improvement for Travel Management and %40 improvement for Loan Application use case as shown in Figures 6.29 and 6.31).

Clustering the process alternatives in multi-organizational environment is also emphasized in process mining literature such that, while hierarchical clustering (agglomerative or divisive) is applied in [93, 94], IR-based multimodal search, DBSCAN and K-Means clustering algorithms are also preferred in [94, 95, 96]. Consequently, Expectation Maximization, Hierarchical Clustering and Simple K-Means are applied with various distance functions (e.g. Manhattan, Eucledian and Minkowski distance functions) at Multi-Reference Pairwise Alignment. According to the clustering results given in Tables 6.19 and

6.20, Expectation Maximization and Simple K-Means algorithms have a better accuracy than Hierarchical Clustering. This result is consistent with [94] such that, K-Means algorithm does not progress the clustering steps upon the prior clustering instances. Hence it results better in clustering in terms of intra- and inter-cluster distance metrics than obtained by hierarchical algorithm.

As the third setting, Multi-Sequence Alignment is implemented in terms of 3 alignment modes: *confidence enhanced sequence alignment* (SA), *sum-of-pairs* and *classical NW.* While classical NW mode just applies the task label similarity search by syntactically comparing the activities in an atomic manner, inDel operation is always penalized by –confThr default value. Because of improper balance between the edit operations scoring, classical NW mode behaves conservative towards matching operation rather than inDel operation. This tendency results in shorter alignments that have *lower behavioral similarity* scores.

Controversially, confidence enhanced SA mode dynamically valuates the edit operations by confidence-aware costing function based on Equations 5.4 - 5.9 such that, the basic idea of this function is to associate the actual frequency of consecutive activity pairs that have common and specific business context with expected frequencies and to interpret whether they occur in a dependent fashion or not. In the case of significant divergences among the business contexts or industry-level requirements, the substitution of contrasting activities are highly penalized, while substitute activities are encouraged to be replaced due the likelihood at business context. On the contrary, insertion or deletion of activities violating the business conditions is highly penalized by inDel operation. Generally, confidence enhanced SA mode tends to prefer matching operation and has relatively *higher structural similarity* scores rather than behavioral similarity. This inhibits the alignment length at moderate lower levels. Alternatively, sum-of-pairs mode is the summation of the scores of all possible pairwise alignments and this costing function is also priorly adapted in process mining literature [21, 62, 84].

In addition to visual analysis based on cluster contents, process family tree frequencies and instances, two distinct metrics are designed for process similarity measurement: *intra*- and *inter-cluster distance*. These metrics interpret the quality of process clustering by measuring how similar the process variants to its neighboring candidates assigned to the same cluster compared to other clusters. Moreover, these two metrics are combined reciprocally within *silhouette measure* that indicates the balance between the cohesion within the cluster and the coupling between the clusters. Respectively, confidence enhanced SA mode builds more compact process clustering with maximal cohesion and minimal coupling rather than sum-of-pairs mode as shown in Figures 6.81 and 6.83.

Consequently, Multi-Reference Pairwise Alignment is consistent with Multi-Sequence Alignment at detecting the *outlier-like* process alternatives. Additionally, sum-of-pairs mode tends to generate hybrid outcomes that combine the tendency of both Multi-Reference Pairwise Alignment and Multi-Sequence Alignment with confidence enhanced SA mode. Moreover, the singleton clustering results of both Multi-Reference Pairwise Alignment and Multi-Sequence Alignment approaches are in parallel with the clustering outcomes obtained at prior studies [27, 28].

The contribution of this study can be summarized as follows:

- Dominant behavior and confidence values provide a log-log similarity measurement which relaxes the requirement for the existence of a reference process model.
- In process mining literature [21, 23, 34, 60, 61, 62], sequence alignment technique has been applied to preprocess the event logs prior to process discovery. In this study, sequence alignment is adapted to measure the degree of similarity among process alternatives.

  Confidence enhanced cost functioning employed at the NW adaptations appropriately eliminates the edit operations that contradict with the underlying business context. While the substitution of contrast activities and inDel operations of activities with little compatibility for the corresponding business rules are avoided dynamically, the tasks with complementary business circumstance are encouraged to be substituted or inserted at practical costs.
- Respectively, prior process similarity measures, i.e. trace equivalence and bisimulation, aims to find out a true/false response rather than the degree of similarity and they handle all components of the corresponding process as equally important [30, 31]. On the contrary, proposed approach measures the degree of similarity on a continuous scale and it checks the balance between the rare active and significant fragments of the process in the context of missfiting.

- As the business value, the alignments of process alternatives that are assigned to the same process family can play a significant role in process configuration such that, conserved regions detected by *maximal identical pairs* (maxIP) with higher frequency and coverage are interpreted as an evidence of common behavior and manifestation of these concurrent behaviors highlight a functional inheritance at process enactment. Consequently, these regions can be refined as the abstractions at the design of configurable process models.
- As the organizations reach higher maturity levels at BPM applications, they tend to accumulate extensive number of reference process models that constitute a valuable asset or intellectual property to business process improvement [17, 18]. Process models are mostly not created from scratch and the duplication of process models is probable. In this aspect, proposed approach can be adapted for process querying to search for the most common business processes and avoid potential redundancies at process modeling.

7.2. Limitations and Future Work

One of the major functionalities of process-aware information systems is to execute the multiple instances of the underlying business process. The motivation of process mining discipline is to discover the process behavior from the runtime information of the process instances, assuming that is possible to record tasks as events and to assign these events to clearly defined process cycles. In other words, it is assumed that the case identifiers exist with process instances in the form of <*caseID, eventID*> event logs. These event logs are formally called *labeled event logs*.

While the logs with automatically assigned case identifiers are classified as maturity level-4 or higher at logging [87], the process can execute in an environment of lower logging maturity level. As stated in [47], the event logs of most of the ERP systems do not allow for monitoring unique and individual process cycles. Instead, they only log the execution of the transactions without referring to the corresponding case. This is due to the fact that, these systems are mostly *data-centric* such that the event logs are staggered at the application tables with the lack of case identifier. Such kind of logs is called as *unlabeled event logs* [1, 8, 47]. To overcome this limitation at the business processes with unlabeled event logs, a program, which generates synthetic events logs for a given reference process structure and the process profile (e.g. activity type and execution probability) according to the Petri-net firing rule, is developed as stated in Section 6.2.2. While synthetic event log generation is applied for only Travel Management use case and logical case identifiers are defined by financial period×plant cartesian at Period End Closing use case, event logs are obtained from process mining repositories for latter use cases.

Another limitation at process similarity measurement is the task label similarity that measures the similarity of the elements in the process model. The similarity between process model elements especially at model-model similarity measurement is calculated from the task labels according to the syntactic similarity measurement, the semantic measurement or the combination of both [17]. While syntactic measurement is based on various algorithms, e.g. string-edit distance, n-gram, morphological analysis and stop-word elimination, semantic techniques are related to the synonym relations captured in thesaurus, e.g. Wordnet [31]. In this study as given in Section 6.2.1, all task labels are unified at a coherent and single activity dictionary for each use case.

Respectively, the potential problem of standard Multi-Sequence Alignment algorithm is the local optimality, which stems from myopic and greedy nature of progressive alignment strategy. This technique combines firstly the closest individuals and the topology of the process family tree is dependent to which individuals are accreted. Indeed, it is NP-hard to get global optimality at root and our ultimate goal is the content of process families at lower cutting levels [84, 85]. On the other hand, the complexity of Multi-Sequence Alignment adaptation in the proposed approach is approximately $O(n^3l^2)$, where n stands for the number of candidate process variants ($|PV|$) and l denotes average length of the sequence (or individual).

Process configuration phase in this study aims to derive common patterns of activity invocations among the process alternatives in the same process family. These commonalities are characterized by the identical pairs introduced by IP and maxIPs. As the future work, it is aimed to extend this study towards automated configurable process model generation such that, the process structures are defined as the least common multiple of all process variants by abstracting the overlapping conserved regions, and the divergence across the process alternatives are dealt with configurable process elements.

REFERENCES

1. Aalst, W. M. P. v. d., Gunther, C., Recker, J., & Reichert, M. (2006). *Using Process Mining to Analyze and Improve Process Flexibility*. BPMDS 2006.

2. Weijters, T. A. J. M. M., & Aalst, W. M. P. v. d. (2003). *Process Mining Discovering Workflow Models from Event-Based Data*. Integrated Computer-Aided Engineering, 10.

3. Maruster, L., Weijters, T. A. J. M. M., Aalst, W. M. P. v. d., & Bosch, A. v. d. (2002). *Process Mining: Discovering Direct Successors in Process Logs*. Lecture Notes in Computer Science, vol. 2534, 364-373.

4. Agraval, R., Gunopulos, D., & Leymann, F. (1998). *Mining Process Models from Workflow Logs*. 6th International Conference on Extending Database Technology.

5. Cook, J. E., & Wolf, A. L. (1996). *Discovering Models of Software Processes from Event-Based Data*. ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 7(3), 215-249.

6. Aalst, W. M. P. v. d., Weijters, T. A. J. M. M., & Maruster, L. (2004). Workflow Mining: Discovering Process Models from Event Logs. Transaction on Knowledge and Data Engineering, vol. 16(9), 1128-1142.

7. Weijters, T. A. J. M. M., & Aalst, W. M. P. v. d. (2003). *Rediscovering Workflow Models from Event-Based Data Using Little Thumb*. Integrated Computer-Aided Engineering, vol. 10(2), 151-162.

8. Gunther, C. W., & Aalst, W. M. P. v. d. (2006). *Process Mining in Case Handling Systems*. Multikonferenz Wirtschaftsinformatik 2006.

9. van Beest, N.R.T.P, Dumas, M., Garcia-Banuelos, L. & La Rosa, M. (2015). *Log Delta Analysis: Interpretable Differencing of Business Process Event Logs*. Business Process Management (BPM), 386-405.

10. Aalst, W. M. P. v. d., Dongen, B. F., Herbst, J., L., M., Schimm, G., & Weijters, T. A. J. M. M. (2003). *Workflow Mining: A Survey of Issues and Approaches*. Data & Knowledge Engineering, vol. 47(2), 237-267.

11. Buijs, J.C., van Dongen, B.F., & van der Aalst, W. M. P. (2012). *Towards Cross-Organizational Process Mining in Collections of Process Models and Their Executions*. Business Process Management Workshops, 2-13.

12. van der Aalst, W.M.P. (2010). *Configurable Services in the Cloud: Supporting Variability While Enabling Cross-Organizational Process Mining*. Confederated International Conferences (CoopIS), vol. 6426, 8-25.

13. van der Aalst, W.M.P. (2011). *Business Process Configuration in the Cloud: How to Support and Analyze Multi-tenant Processes*. 9th IEEE European Conference on Web Services (ECOWS), 3-10.

14. van der Aalst, W. M. P., Lohmann, N., La Rosa, M., & Xu, J. (2010). *Correctness Ensuring Process Configuration: An Approach Based on Partner Synthesis*. 8th International Conference on Business Process Management (BPM), vol. 6336, 95-111.

15. van der Aalst, W. M. P. (2011). *Intra and Inter-Organizational Process Mining: Discovering Processes within and between Organizations*. IFIP Working Conference on the Practice of Enterprise Modeling, vol. 92, 1-11.

16. Gottschalk, F., van der Aalst, W.M.P. & Jansen-Vullers, M. H. (2007). *Configurable Process Models: A Foundational Approach*. Reference Modeling, 59-77.

17. Dumas, M., Garcia-Banuelos, L., & Dijkman, R. (2009). *Similarity Search of Business Process Models*. IEEE Data Engineering Bulletin, vol. 32(3), 23-28.

18. Wang, J., He, T., Wen, L., Wu, N., Hofstede, A. H. M., & Su, J. (2010). *A Behavioral Similarity Measure between Labeled Petri Nets Based on Principal Transition Sequences*. On the Move to Meaningful Internet Systems: OTM 2010, Confederated International Conferences: CoopIS, IS, DOA and ODBASE, vol. 6426, 394-401.

19. Becker, M., & Laue, R. (2012). *A Comparative Survey of Business Process Similarity Measures*. Computers in Industry, vol. 63(2), 148-167.

20. De Medeiros, A. K. A., van der Aalst, W. M. P., & Weijters, A. J. M. M. (2008). *Quantifying Process Equivalence Based on Observed Behavior*. Data and Knowledge Engineering, vol. 64(1), 55-74.

21. Jagadeesh Chandra Bose, R. P., & van der Aalst, W. M. P. (2010). *Trace Alignment in Process Mining: Opportunities for Process Diagnostics*. 8th International Conference, Business Process Management (BPM), vol. 6336, 227-242.

22. Sung, W. (2010) *Algorithms in Bioinformatics: A Practical Introduction*. Chapman and Hall/CRC Press.

23. Jagadeesh Chandra Bose, R. P., & van der Aalst, W. M. P. (2012). *When Process Mining Meets Bioinformatics*. CAISE Forum, IS Olympics: Information Systems in a Diverse World, vol. 107, 202-217.

24. Maruster, L., Weijters, T., & Bosch, A. v. d. (2006). *A Rule-Based Approach for Process Discovery*. Data Mining and Knowledge Discovery, vol. 13(1), 67-87.

25. Aalst, W. M. P. v. d., Rubin, V., Dongen, B. F., & Gunther, C. W. (2007). *Process Mining: A Two-Step Approach Using Transition Systems and Regions*. Business Process Management, vol. 4714, 375-383.

26. Gomez, J. M., Kassem, G., Rauntenstrauch, C., & Melato, M. (2003). *Analysis of User's Behaviour in Very Large Business Application Systems with Methods of the Web Usage Mining- A Case Study on* SAP® R/3®. Advance in Web Intelligence, vol. 2663, 954-964.

27. Buijs, J.C., & Reijers, H.A. (2014). *Comparing Business Process Variants Using Models and Event Logs. Enterprise*, Business-Process and Information Systems Modeling, 154-168.

28. Yilmaz, O., & Karagoz, P. (2015). *Generating Performance Improvement Suggestions by using Cross-Organizational Process Mining*. 5th International Symposium on Data-Driven Process Discovery and Analysis, 3-17.

29. Rozinat, A., & van der Aalst, W. M. P. (2006). *Conformance Testing: Meausuring the Fit and Appropriateness of Event Logs and Process Models*. Business Process Management Workshops (BPM), vol. 3812, 163-176.

30. van Dongen, B., Dijkman, R., & Mendling, J. (2008). *Measuring Similarity between Business Process Models*. Advanced Information Systems Engineering, 20th International Conference, CAISE, vol. 5074, 450-464.

31. Mendling, J., van Dongen, B., & van der Aalst, W. M. P. (2007). *On the Degree of Behavioral Similarity between Business Process Models*. 6th GI Workshop on Event-Driven Process Chains, CEUR-WS.org, vol. 303, 39-58.

32. Dijkman, R. M., van Dongen, B. F., Dumas, M., Garcia-Banuelos, L., Kunze, M., Leopolod, H., Mendling, J., Uba, R., Weidlich, M., Weske, M., & Yan, Z. (2013). *A Short Survey on Process Model Similarity*. Seminal Contributions to Information Systems Engineering, 421-427.

33. Jagadeesh Chandra Bose, R. P., & van der Aalst, W. M. P. (2009). *Abstractions in Process Mining: A Taxonomy of Patterns*. International Workshops, Business Process Management (BPM), vol. 5701, 159-175.

34. Song, M., Gunther, C. W., & van der Aalst, W. M. P. (2009). *Trace Clustering in Process Mining*. International Workshop, Business Process Management (BPM), vol. 17, 109-120.

35. Aalst, W. M. P. v. d. (2006). *Workshop Report: Process Mining, Monitoring Processes and Services*. Paper presented at the Dagstuhl Seminar Proceedings, Dagstuhl, Germany.

36. Weijters, T. A. J. M. M., Aalst, W. M. P. v. d., & Medeiros, A. K. A. (2006). *Process Mining with the HeuristicMiner Algorithm*. Paper presented at the BETA Working Paper Series, WP 166,, Eindhoven University of Technology.

37. Maruster, L., Aalst, W. M. P. v. d., Weijters, T. A. J. M. M., Bosch, A. v. d., & Daelemans, W. (2001). *Automated Discovery of Workflow Models for Hospital Data*. Paper presented at the Proceeding BNAIC-01.

38. Aalst, W. M. P. v. d., Reijers, H. A., Weijters, T. A. J. M. M., van Dongen, B. F., Song, M., & Verbeek, H. M. W. (2007). *Business Process Mining: An Industrial Application*. Information Systems, vol. 32(5), 713-732.

39. Chiravalloti, A. D., Greco, G., Guzzo, A., & Pontieri, L. (2006). *An Information-Theoretic Framework for Process Structure and Data Mining*. Data Warehousing and Knowledge Discovery, vol. 4081, 248-259.

40. Greco, G., Guzzo, A., Manco, G., & Sacca, D. (2007). *Mining Unconnected Patterns in Workflows*. Information Systems, vol. 32(5), 685-712.

41. Greco, G., Guzzo, A., Luigi, P., & Sacca, D. (2004). *Mining Expressive Process Models by Clustering Workflow Traces*. PAKDD 2004, Heidlberg.

42. Esgin, E., & Senkul, P. (2009). *Hybrid Approach to Process Mining: Finding Immediate Successors of a Process by Using From-to Chart*. International Conference on Machine Learning and Applications, 664-668.

43. Esgin, E., Senkul, P., & Cimenbicer, C. (2010). *A Hybrid Approach for Process Mining: Using From-to Chart Arranged by Genetic Algorithms*. Hybrid Artifical Intelligence Systems, vol. 6076, 178-186.

44. Esgin, E., & Senkul, P. (2011). *Extracting Connection Types in Process Models Discovered by Using From-to Chart Based Approach*. Developing Concepts in Applied Intelligence. Studies in Computational Intelligence, vol. 363, 59-65.

45. Dijkman, R., Dumas, M., van Dongen, B., Kaarik, R., & Mendling, J. (2011). *Evaluating Distance Functions for Clustering Tandem Repeats*. Information Systems, vol. 36, 498-516.

46. Breugel, F. V., & Worrel, J. (2005). *A Behavioural Pseudometric for Probabilistic Transition Systems*. Theoretical Computer Science, vol. 331(1), 115-142.

47. van der Aalst, W. M. P. (2006). *Matching Observed Behavior and Modeled Behavior: An Approach Based on Petri nets and Integer Programming*. Decision Support Systems, vol. 42(3), 1843-1859.

48. Kunze, M., Weidlich, M., & Weske, M. (2011). *Behavioral Similarity: A Proper Metric*. 9th International Conference, Business Process Management (BPM), vol. 6896, 166-181.

49. Zha, H., Wang, J., Wen, L., Wang, C., & Sun, J. (2010). *A Workflow Net Similarity Measure Based on Transition Adjacency Relations*. Computers in Industry, vol. 61, 463-471.

50. Gerke, K., Cardoso, J., & Claus, A. (2009). *Measuring the Compliance of Processes with Reference Models*. On the Move to Meaningful Internet Systems: OTM 2009, Confederated International Conferences: CoopIS, IS, DOA and ODBASE, vol. 5870, 76-93.

51. Roseman, M. (2006). *Potential Pitfalls of Process Modeling*. Business Process Management, vol. 12(2), 249-254.

52. Vanderfeesten, I. T. P., Cardoso, J., Mendling, J., Reijters, H. A., & van der Aalst, W. M. P. (2007). *Quality Metrics for Business Process Models*. Business Process Management and Workflow Handbook, 179-190.

53. Rozinat, A., & van der Aalst, W. M. P. (2008). *Evaluating the Quality of Discovered Process Models*. 2nd International Workshop on the Induction of Process Models (IPM), 45-52.

54. Stolfa, J., Kopka, M., Slaninova, K., & Martinovic, J. (2014). *An Application of Process Mining by Sequence Alignment Methods to the SAP Invoice Process Example*. Advances in Intelligent Systems and Computing, vol. 237, 61-74.

55. Gao, J., Zhao, D., Zhou, H., & Wang, Y. (2014). *Alignment-based Similarity Measurement for Action Traces*. International Journal of Computer Science and Network Security, vol. 14(8), 30-35.

56. Juan, Y. C. (2006). *A String Comparison Approach to Process Logic Differences between Business Process Models*. 9th Joint Conference on Information Sciences (JCIS), Advances in Intelligence Systems Research.

57. Rao, S., Rodriquez, A., & Benson, G. (2005). *Evaluating Distance Functions for Clustering Tandem Repeats*. Genom Informatics, vol. 16(1), 3-12.

58. Jagadeesh Chandra Bose, R. P., & van der Aalst, W. M. P. (2010). *Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models*. 7th International Conference, Business Process Management (BPM), vol. 43, 170-181.

59. De Weerdt, J., Broucke, S. v., Vanthienen, J., & Baesens, B. (2013). *Active Trace Clustering for Improved Process Discovery*. Knowledge and Data Engineering, IEEE Transactions vol. 25(12), 2708-2720.

60. Veiga, G. M., & Ferreira, D. R. (2010). *Understanding Spaghetti Models with Sequence Clustering for ProM*. International Workshops, Business Process Management (BPM), 92-103.

61. Evermann, J., Thaler, T., & Fettke, P. (2015). *Clustering Traces Using Sequence Alignment*. International Workshops, Business Process Management (BPM).

62. Jagadeesh Chandra Bose, R. P., & van der Aalst, W. M. P. (2009). *Context Aware Trace Clustering: Towards Improving Process Mining Results*. SIAM International Conference on Data Mining (SDM), 401-412.

63. Lesh, N., Zaki, M. J., & Ogihara, M. (1999). *Mining Features for Sequence Classification*. Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 342-346.

64. Mendling, J., & Strembeck, M. (2008). *Influence Factors of Understanding Business Process Models*. 11th International Conference, Business Information Systems (BIS), vol. 7, 142-153.

65. Esgin, E., & Senkul, P. (2011). *Delta Analysis: A Hybrid Quantitative Approach for Measuring Discrepancies between Business Process Models*. 6th International Conference, Hybrid Artificial Intelligence Systems (HAIS), vol. 6678, 296-304.

66. Esgin, E., & Karagoz, P. (2013). *Sequence Alignment Adaptation for Process Diagnostics and Delta Analysis*. 8th International Conference, Hybrid Artificial Intelligence Systems (HAIS), vol. 8073, 191-201.

67. Esgin, E., & Karagoz, P. (2013). *Confidence-Aware Sequence Alignment for Process Diagnostics*. International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 990-997.

68. Esgin, E., & Karagoz, P. (2015). *Dynamic Scoring-based Sequence Alignment for Process Diagnostics*. Current Approaches in Applied Artificial Intelligence, vol. 9101, 742-752.

69. Adriansyah, A., van Dongen, B. F., & van der Aalst, W. M. P. (2011). *Conformance Checking using Cost-Based Fitness Analysis*. IEEE 15th International Enterprise Distributed Object Computing Conference (EDOC), 55-64.

70. Nguyen, H., Dumas, M., La Rosa, M., Maggi, F.M., & Suriadi, S. (2014). *Mining Business Process Deviance: A Quest for Accuracy*. Confederated International Conferences (CoopIS), vol. 8841, 436-445.

71. Apple, J. M. (1972). *Material Handling Systems Design*. New York: The Ronald Press Company.

72. Francis, R. L., McGinnis L. F., & White, J. A. (1992). *Facility Layout and Location: An Analytical Approach*. New Jersey: Prentice Hall.

73. Meyers, F. E., & Stephens, M. P. (2005). *Manufacturing Facilities Design and Material Handling*. New Jersey: Pearson Prentice Hall.

74. Witten, H. I., & Frank, E. (2000). *Data Mining Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Series in Data Management Systems.

75. Winston, W. L., & Goldberg, J. B. (2004). *Operations Research: Applications and Algorithms*: Thomson Delmar Learning.

76. Barricelli, N. A. (1963). *Numerical Testing of Evolution Theories*. Part II. Preliminary Tests of Performance, Symbiogenesis and Terrestrial Life. Acta Biotheoretica vol. (16): 99-126.

77. Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall.

78. Dianati, M., Song, I., & Treiber, M. (2010). *An Introduction to Genetic Algorithms and Evaluation Strategies*. University of Waterloo, Canada.

79. Beasley, D., Bull, D. R., & Martin, R. R. (1993). *An Overview of Genetic Algorithms: Part 1*, Fundamentals. University Computing, vol. 15(2), 58-69.

80. Affenzeller, M., Wagner, S., Winkler S., & Beham, A. (2009). *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Florida : CRC Press.

81. Aalst, W. M. P. v. d. (2005). *Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing*. Requirement Engineering, vol. 10(3). 198-211.

82. Aalst, W. M. P. v. d., Mederios, A., & Weijters, A.J.M.M. (2006). *Process Equivalence: Comparing Two Process Models Based on Observed Behavior*. International Conference on Business Process Management, vol. 4102. 129-144.

83. Esgin, E. (2009). *A Hybrid Quantitative Approach in Process Modeling: From-to Chart Based Process Discovery*. Master of Science Dissertation. METU Ankara.

84. Thompson, J. D., Higgins, D. G., & Gibson, T. J. (1994) *CLUSTAL W: Improving The Sensitivity of Progressive Multiple Sequence Alignment through Sequence Weighting, Positions-specific Penalties and Weight Matrix Choice.* Nucleic Acids Research, vol. 22, 4673-4680.

85. Feng, D. F., & Doolittle, R. F. (1987) *Progressive Sequence Alignment as a Prerequisite To Correct Phylogenetic Trees*. Journal of Molecular Evolution, vol. 25(4), 351-360.

86. Phillips, A., Janies, D., & Wheeler, W. (2000) *Multiple Sequence Alignment in Phylogenetic Analysis*, Molecular Phylogenetic Evolution, vol. 16(3), 317-330.

87. Bayomie, D., Helal, I. M. A., Awad, A., Ezat, E., & Elbastawissi, A. (2016). *Deducing Case IDs for Unlabeled Event Logs*. International Conference on Business Process Management, vol. 256, 242-254.

88. Blevi, L., Delporte, L., & Robbrecht, J. (2017) *Process Mining on The Loan Application Process of a Dutch Financial Institute*. BPI Challenge 2017.

89. Rozinat, A., & van der Aalst, W. M. P. (2008) *Conformance Checking of Processes Based on Monitoring Real Behavior*. Information Systems, vol. 33(1), 64-95.

90. Buijs, J. Environmental Permit Application Process (WABO), CoSeLoG Project (2014), http://dx.doi.org/10.4121/uuid:26aba40d-8b2d-435b-b5af-6d4bfbd7a270.

91. Dijkman, R., Dumas, M., van Dongen, B., Kaarik, R., & Mendling, J. (2011) *Similarity of Business Process Models: Metrics and Evaluation*. Information Systems, vol. 36, 498-516.

92. Dongen, S., & Enright A. J. (2012) *Metric Distances Derived From Cosine Similarity and Pearson and Spearman Correlations.* CoRR,

93. Jung, J., Bae, J., & Liu, L. (2009) *Hierarchical Clustering of Business Process Models.* International Journal of Innovative Computing, Information & Control vol. 5(12), 4501-4511.

94. Melcher, J., & Seese, D. (2008) *Visualization and Clustering of Business Process Collections Based on Process Metric Values.* 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing.

95. Ordonez, H., Merchan, L., Ordonez, A., & Cobos, C. (2016) *Business Process Models Clustering Based on Multimodal Search, K-means, and Cumulative and No-Continuous N-Grams*. Polibits, 25-31.

96. Francescomarino, C., Dumas, M., Maggi, F. M., & Teinemaa, I. (2016) *Clustering-Based Predictive Process Monitoring*. IEEE Transaction on Services Computing PP(99).

97. Chuong, B. D., & Katoh, K. (2008) *Protein Multiple Sequence Alignment*. Methods in Molecular Biology. vol. 484, 379-413

98. Carrillo, H., & Lipman, D. (1988) *The Multiple Sequence Alignment Problem in Biology*. SIAM Journal on Applied Mathematics. vol. 48(5), 1073-1082.

99. Kececioglu, J., & Starrett, D. (2004) *Aligning Alignments Exactly*. The 8th Annual İnternational Conference on Resaerch in Computational Molecular Biology. 85-96.

100. Gudivada, V. N., Rao, D. L., & Gudivada, A. R. (2018) *Information Retrieval: Concepts, Models and Systems*. Handbook of Statistics. vol. 38, 331-401.

101. Aysolmaz, B., Schunselaar, D. M. M., Reijers, H. A., & Yaldiz, A. (2017). *Selecting A Process Variant Modeling Approach: Guidelines And Application*. Software & Systems Modeling. 1-24.

APPENDICES

APPENDIX A – ConfEnhMSA User Guide

Confidence Enhanced Multi-Sequence Alignment (aka. confEnhMSA) is the program that utilizes an adaptation of NW algorithm iteratively to achieve the multi-sequence alignment of a set of dominant behaviors. As the outcomes, confEnhMSA constructs the process family trees that depict the relative likelihood among the candidate process alternatives. Additionally, all valid alignments and NW tables are given in detailed.

Technically, confEnhMSA is developed in Java and constitutes of following classes as given in Figure A.1.

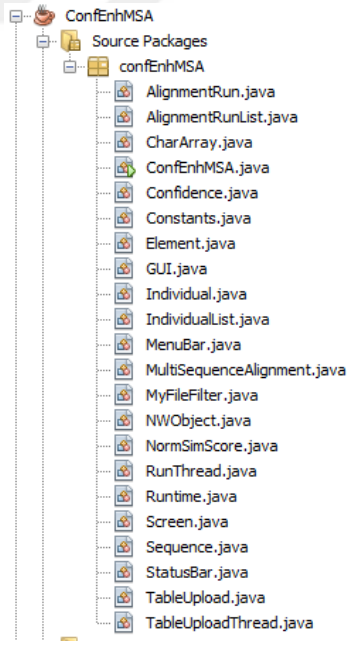


Figure A.1. List of Classes for confEngMSA Program.

Figure A.2 shows the user interface at the initialization of confEnhMSA program.

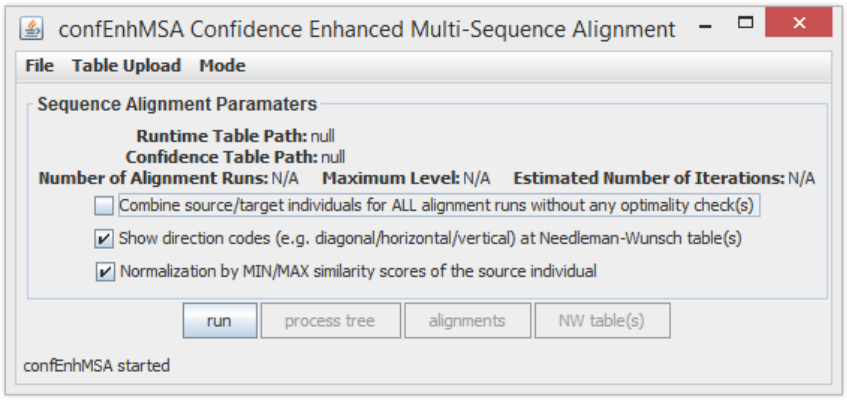


Figure A.2. User Interface for confEnhMSA Program.

1. Table Upload

Prior to the execution of multi-sequence alignment, two input tables should be uploaded: *runtime* and *confidence tables*.

- Runtime table holds the alignment runID (i.e. a code for uniquely identifying a single alignment run), relevant confidence tableID assigned to the alignment run, the confidence threshold (*confThr*) and the consensus activity sequences that hold the dominant behavior per process alternative.
- Confidence table holds the confidence values of the predecessor/successor activity pairs per confidence tableID.

Sample formats for runtime and confidence table are given in Figures A.3 and A.4 respectively.

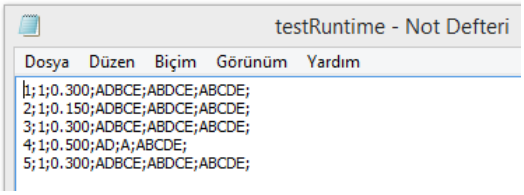


Figure A3. Sample Format of Runtime Table.

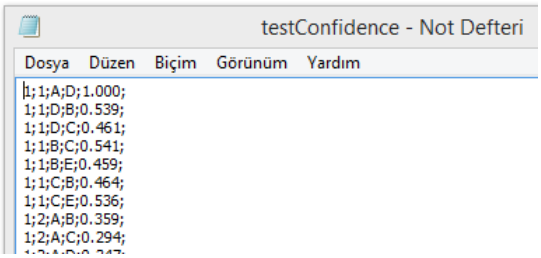


Figure A.4. Sample Format of Confidence Table.

These input tables can be uploaded via Table Upload menu bar item as shown in Figure A5. After uploading the runtime and confidence tables, the paths of the underlying files, number of alignment runs, the maximum level for process trees and estimated number of iterations fields at the user interface are filled automatically.

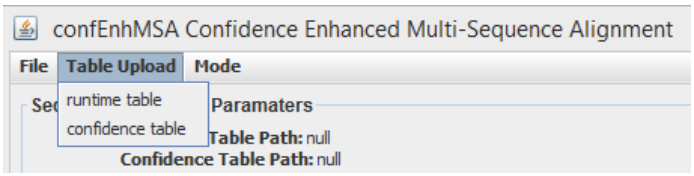


Figure A.5. Table Upload Menu Bar Item Selection.

2. Additional Configurations

Additional configurations prior to multi-sequence alignment execution are as follows:

- Mode setting. One of the confEnhMSA functionalities is the confidence-enhanced cost functioning which avoid the edit operations that do not make sense according to business context such that; the substitution of uncorrelated, contrasting activities or indel operations (insertion/deletion) of activities not conforming to the business rules should be penalized, while complementary tasks should be encouraged to be replaced or inserted at sensible costs.

  Additionally, classical NW pay-off matrices for extra what-if analysis (i.e. +*confThr* for matching, -*confThr* for mismatching and indel edit operations) can be applied. Mode setting can be configured from Mode menu bar item as shown Figure A.6.

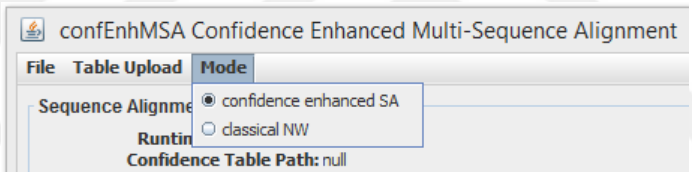


Figure A.6. Mode Menu Bar Item Selection.

- Combine source and target individuals for all alignments. In the case of optimality for the current level, a combined individual (i.e. the compound of aligned forms of source and target individuals) is created and passed to the next level. But this combined individual creation can be performed for all alignments without any optimality check.
- Show direction codes at NW table. As one of the outcomes, NW table for each alignment is delisted. At these tables, the backtracking directions (horizontal, vertical and diagonal) can be denoted.
- Normalization by MIN/MAX similarity scores of the source individual. After all alignments at the current level are completed, the similarity scores are normalized according to the minimum/maximum (*MIN*/*MAX*) similarity scores of source individual. As an alternative, these similarity scores can be directly evaluated without any normalization for additional what-if analysis.

After completing the additional configurations, Run button should be pressed to execute the multi-sequence alignment.

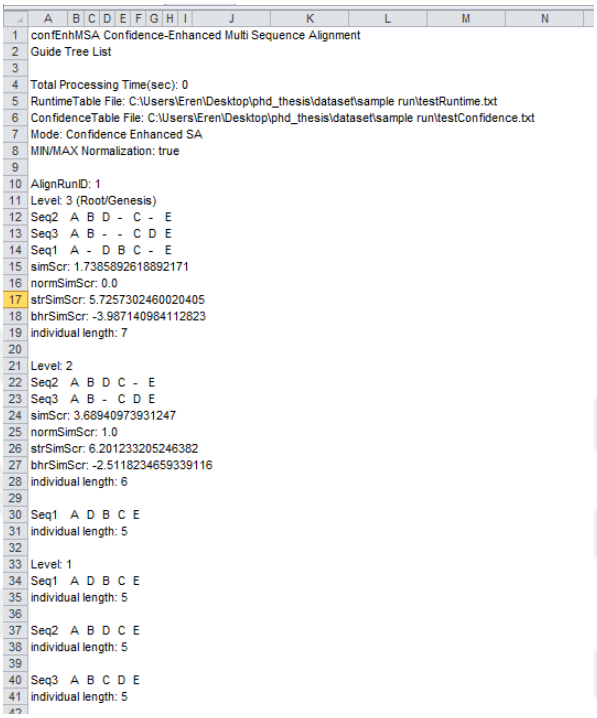
3. The Output Tables

There are majorly three output tables in csv format as follows:

i. Process tree. All candidate individuals and the optimal individual with its similarity scores are listed from the root node to the base level.

ii. Alignment run list. All alignments with source, target individuals and similarity scores are given in alignment runID and level basis.
iii. NW table list. The structures of NW tables per alignment are depicted with/without backtracking directions.

Figures A.7 - A.9 show the sample file contents respectively.



confEnhMSA Confidence-Enhanced Multi Sequence Alignment
Guide Tree List

Total Processing Time(sec): 0
RuntimeTable File: C:\Users\Eren\Desktop\phd_thesis\dataset\sample run\testRuntime.txt
ConfidenceTable File: C:\Users\Eren\Desktop\phd_thesis\dataset\sample run\testConfidence.txt
Mode: Confidence Enhanced SA
MIN/MAX Normalization: true

AlignRunID: 1
Level: 3 (Root/Genesis)
Seq2 A B D - C - E
Seq3 A B - - C D E
Seq1 A - D B C - E
simScr: 1.7385892618892171
normSimScr: 0.0
strSimScr: 5.7257302460020405
bhrSimScr: -3.987140984112823
individual length: 7

Level: 2
Seq2 A B D C - E
Seq3 A B - C D E
simScr: 3.68940973931247
normSimScr: 1.0
strSimScr: 6.201233205246382
bhrSimScr: -2.5118234659339116
individual length: 6

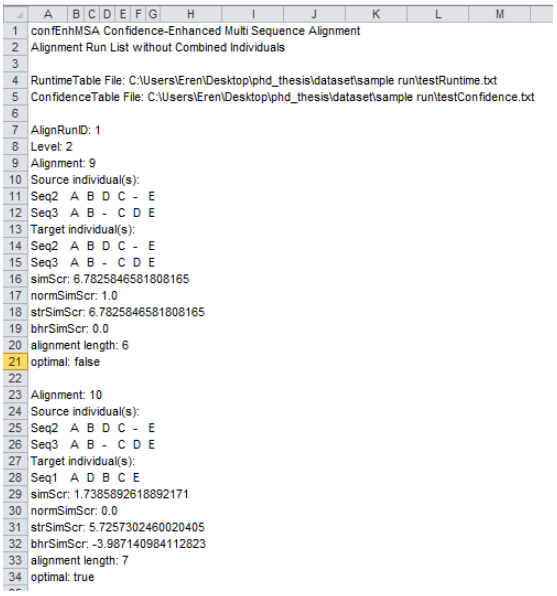
Seq1 A D B C E
individual length: 5

Level: 1
Seq1 A D B C E
individual length: 5

Seq2 A B D C E
individual length: 5

Seq3 A B C D E
individual length: 5

Figure A.7. Sample Content for Process Tree Output.



confEnhMSA Confidence-Enhanced Multi Sequence Alignment
Alignment Run List without Combined Individuals

RuntimeTable File: C:\Users\Eren\Desktop\phd_thesis\dataset\sample run\testRuntime.txt
ConfidenceTable File: C:\Users\Eren\Desktop\phd_thesis\dataset\sample run\testConfidence.txt

AlignRunID: 1
Level: 2
Alignment: 9
Source individual(s):
Seq2 A B D C - E
Seq3 A B - C D E
Target individual(s):
Seq2 A B D C - E
Seq3 A B - C D E
simScr: 6.7825846581808165
normSimScr: 1.0
strSimScr: 6.7825846581808165
bhrSimScr: 0.0
alignment length: 6
optimal: false

Alignment: 10
Source individual(s):
Seq2 A B D C - E
Seq3 A B - C D E
Target individual(s):
Seq1 A D B C E
simScr: 1.7385892618892171
normSimScr: 0.0
strSimScr: 5.7257302460020405
bhrSimScr: -3.987140984112823
alignment length: 7
optimal: true

Figure A.8. Sample Content for Alignment Run List Output.

confEnhMSA Confidence-Enhanced Multi Sequence Alignment
Needleman-Wunsch Tables (F)

RuntimeTable File: C:\Users\Eren\Desktop\phd_thesis\dataset\sample run\testRuntime.txt
ConfidenceTable File: C:\Users\Eren\Desktop\phd_thesis\dataset\sample run\testConfidence.txt

AlignRunID: 1
Level: 2

Alignment: 10
optimal: true
source individual: Seq2 Seq3
target individual: Seq1

		A	D	B	C	E
	0.000 (r)	-0.300 (h)	-0.600 (h)	-0.900 (h)	-1.200 (h)	-1.500 (h)
A	A -0.300 (v)	3.035 (d)	0.683 (h)	-1.195 (h)	-2.532 (h)	-1.800 (v)
B	B -0.600 (v)	1.391 (v)	0.710 (d)	1.477 (d)	1.826 (h)	0.851 (h)
D	- -0.900 (v)	0.687 (v)	2.158 (d)	1.725 (h)	1.122 (v)	1.497 (v)
C	C -1.200 (v)	-1.638 (v)	0.962 (v)	0.537 (d)	3.349 (d)	2.544 (h)
-	D -1.500 (v)	-3.150 (h)	0.458 (d)	-0.793 (h)	1.439 (v)	1.984 (v)
E	E -1.800 (v)	-2.100 (h)	-1.192 (v)	-1.720 (h)	-0.211 (v)	1.739 (d)

Level: 1

Alignment: 5
optimal: true
source individual: Seq2
target individual: Seq3

	A	B	C	D	E
0.000 (r)	-0.300 (h)	-0.600 (h)	-0.900 (h)	-1.200 (h)	-1.500 (h)
A -0.300 (v)	2.296 (d)	-0.704 (h)	-2.370 (d)	-2.579 (d)	-1.800 (v)
B -0.600 (v)	0.658 (v)	4.470 (d)	2.928 (h)	0.615 (h)	-1.035 (h)
D -0.900 (v)	-0.749 (v)	3.064 (v)	1.521 (v)	5.053 (d)	3.403 (h)
C -1.200 (v)	-2.425 (v)	1.387 (v)	4.494 (d)	3.389 (h)	3.077 (v)
E -1.500 (v)	-1.800 (h)	-0.263 (v)	4.536 (v)	3.573 (h)	3.689 (d)

Figure A.9. Sample Content for NW Table Output.

APPENDIX B – Extra Process Discovery Analysis

While completeness measures the fraction of the traces in the event log that have some correspondence at the discovered process model, soundness is related to the simplicity and minimality which is also emphasized by Occam's Razor. According to Figure B.1 for Loan Application use-case, process alternatives *cand2*, *cand3* and *cand4* have approximately 100% completeness values, which indicate high accuracy at extracting the dominant behavior. On the other hand, dominant behavior is partially discovered for process alternatives *reference* and *cand1*. This is due to the spaghetti-like process structure dominated with higher connectivity and lower density ratios given in Table B.1.

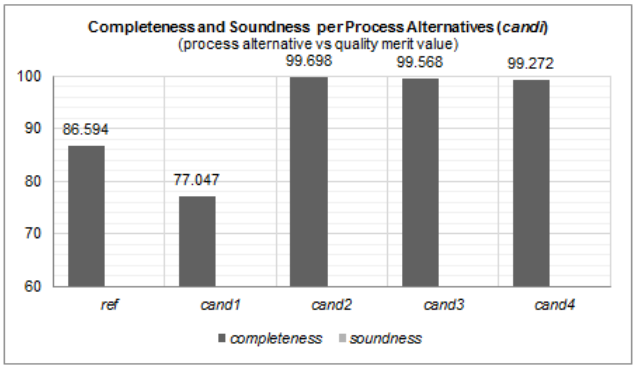


Figure B.1. Average Completeness and Soundness Values per Process Alternatives (*candi*) for Loan Application Use Case.

Table B.1. Structural Factors per Process Alternatives for Loan Application Use Case.

candi	Number of Activities	Number of Transitions	Number of Connectors	Connectivity	Density	Average Transition Length
ref	8	14	5	1,75	0,62	1,59
cand1	8	17	6	2,13	0,57	1,56
cand2	8	10	2	1,25	0,80	1,24
cand3	6	6	1	1,00	0,86	1,19
cand4	8	11	3	1,38	0,73	1,37

Respectively, weak-order transitions at process alternatives *reference* and *cand1* are more tend to be eliminated by the increase at the confidence and support threshold. Hence this pruning down diminishes the coverage of the dominant behavior and completeness. As shown in Figure B.2, the AND/OR/XOR gateways and weak-order transitions initiated by these gateways are effected by the change at [0.3, 0.5] confidence interval. On the contrary, the process behaviors at structured (*lasagna-like*) process alternatives (i.e. *cand2*, *cand3* and *cand4*) are more robust to these changes at confidence threshold.

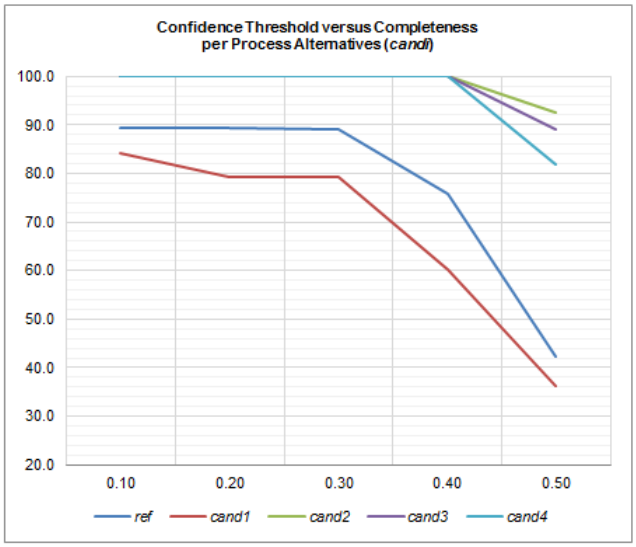


Figure B.2. Confidence Threshold versus Completeness per Process Alternatives (*candi*) for Loan Application Use Case. Respectively more *lasagna-like* process alternatives (e.g. *cand2, cand3* and *cand4*) are more robust to the changes at confidence threshold such that; *lower connectivity* or *higher density* refers to *direct-successor type* transitions and these process behaviors resist to the increase of confidence threshold value.

As seen in Figure B.3, dominant behavior runs mostly capture the behavior at a certain degree, having either completeness over 75% for Environmental Permit Application Use-Case. Respectively, for process alternatives *wabo2 wabo5* only some of the behavior can be captured by dominant behavior. This is possibly due to high variation at the event logs due to the XOR-gateways.

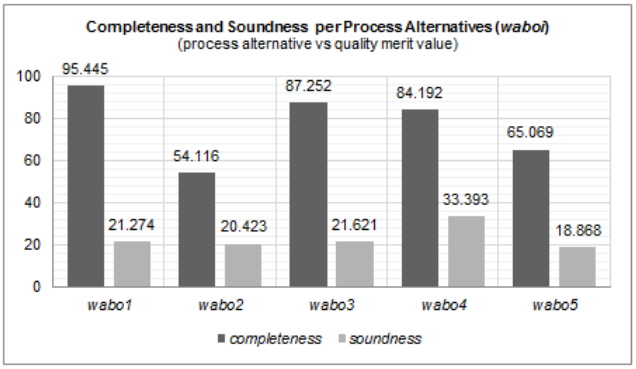


Figure B.3. Average Completeness and Soundness Values per Process Alternatives (*waboi*) for Environmental Permit Application Use Case.

Figure B.4 visualizes the completeness and soundness values on runtime basis. The process alternatives *wabo1* and *wabo3* that are grouped at the upper left-hand side of the figure show relatively a good balance between completeness and soundness quality metrics. On the other hand, the distribution of *wabo4* indicates an overfitting, which generates a highly-specific process model explaining a particular sample event log.

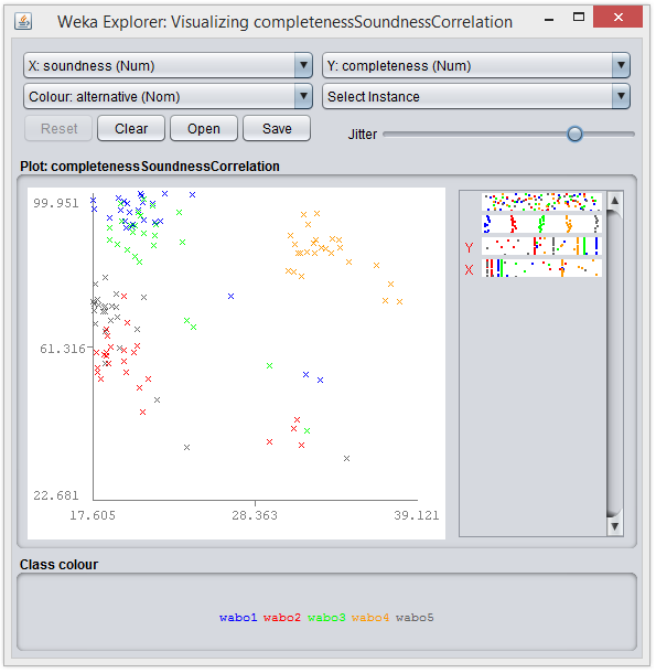


Figure B.4. Average Soundness and Completeness Values per Process Alternatives (*waboi*) for Environmental Permit Application Use Case.

Appendix C

Appendix C

Appendix C

Appendix C

Appendix D

Appendix D

APPENDIX E1 – Cosine Similarity Transformation Details

According to the cosine similarity scores obtained for Environmental Permit Application use-case, it is clear that process alternative *wabo4* is quite distinct from other alternatives according to the discrepancies at Figure E1.4. Hence this process variant can be handled as a *singleton*, i.e. a process cluster consisting of a single process variant. In the case of *numbCluster=3*, visually segregation of other process alternatives is not so practical. According to the gap between the primary trend, which is mostly positioned at 1.0 cosine similarity (exactly the same) level and the secondary (and following) trends, process alternative *wabo1* may be assigned to a distinct cluster. Figures E1.1 - E1.5 depict the cosine similarity transformation.

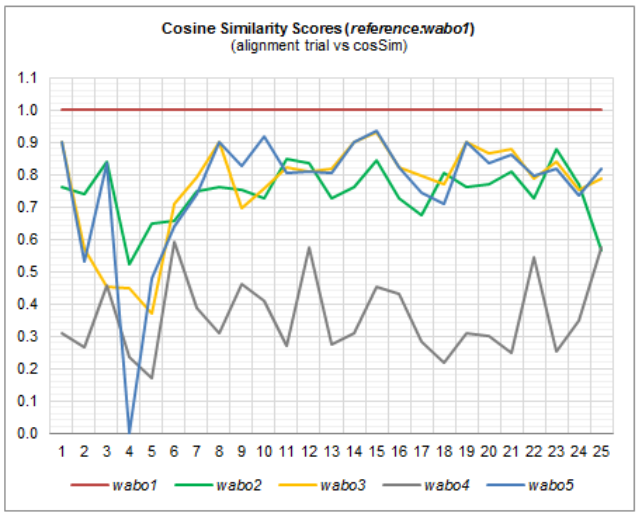


Figure E1.1. Cosine Similarity Score (*cosSim*) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo1*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

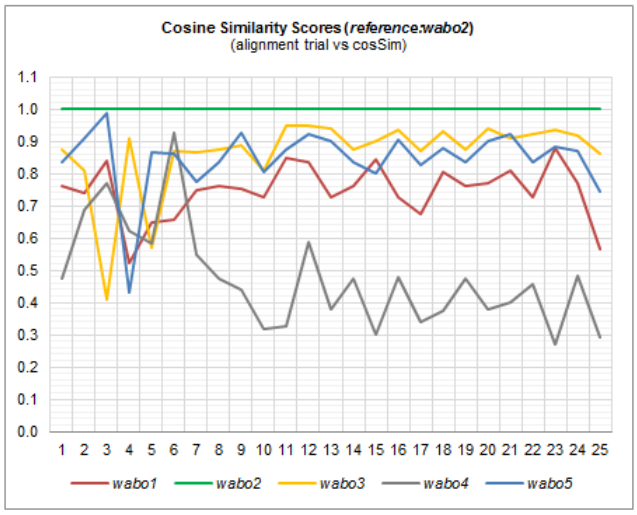


Figure E1.2. Cosine Similarity Score (cosSim) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo2*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

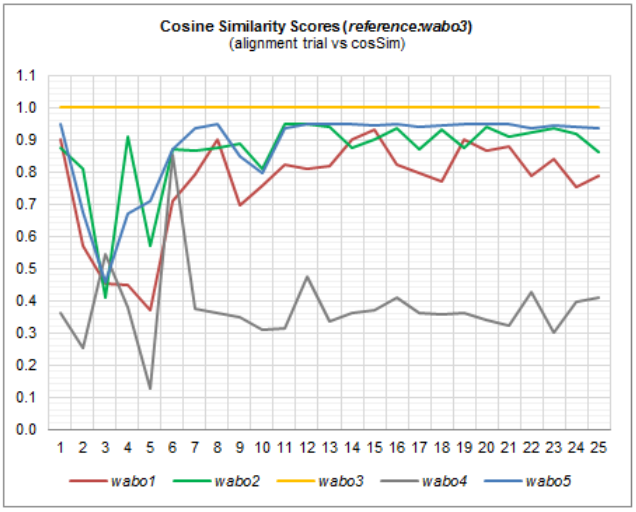


Figure E1.3. Cosine Similarity Score (cosSim) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo3*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

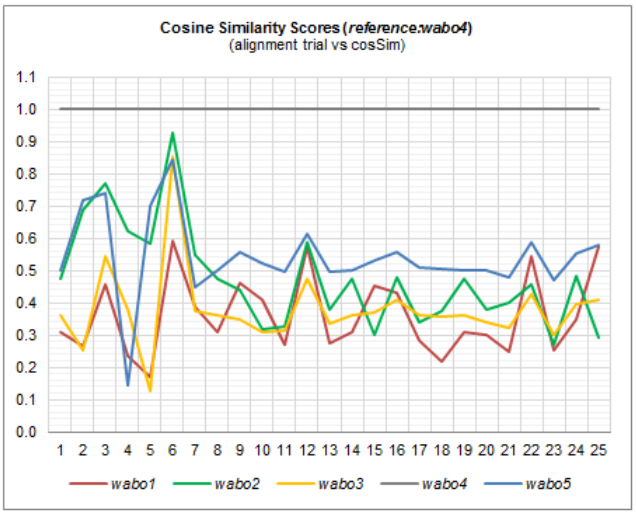


Figure E1.4. Cosine Similarity Score (cosSim) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo4*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

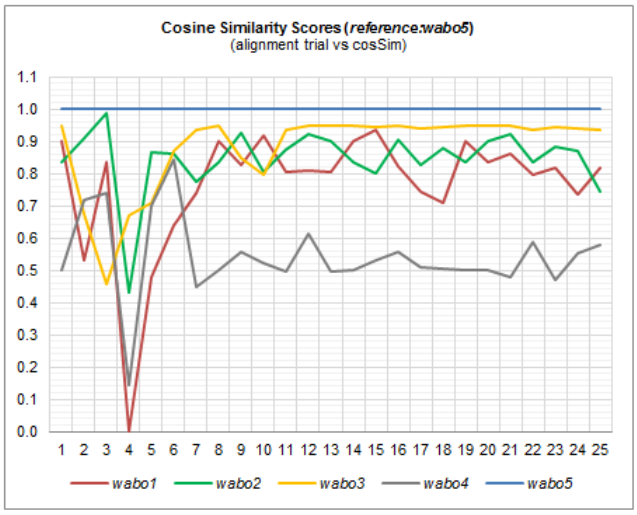


Figure E1.5. Cosine Similarity Score (cosSim) per Pairwise Alignment Run for Environmental Permit Application Use Case (*reference:wabo5*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

According to the cosine similarity scores for Period End Closing use-case, the corresponding process alternatives are exactly segregated into two exact clusters: {*wabo2, wabo4*} and {*wabo1, wabo3, wabo5*}. As shown at Figures E1.7 and E1.9, the underlying cosine similarity score of 0.5 emphasizes a loosely matching between process alternatives *client2* and *client4* and the lack of some Product Costing and Material Ledger functionalities may result in such a posterior neighborhood. On the other hand, process alternatives *client1*, *client3* and *client5* have a relatively stronger likelihood with an average cosine similarity score of 0.649. Especially, the correlation between process alternatives *client1* and *client3* may establish an earlier convergence at the corresponding process cluster.

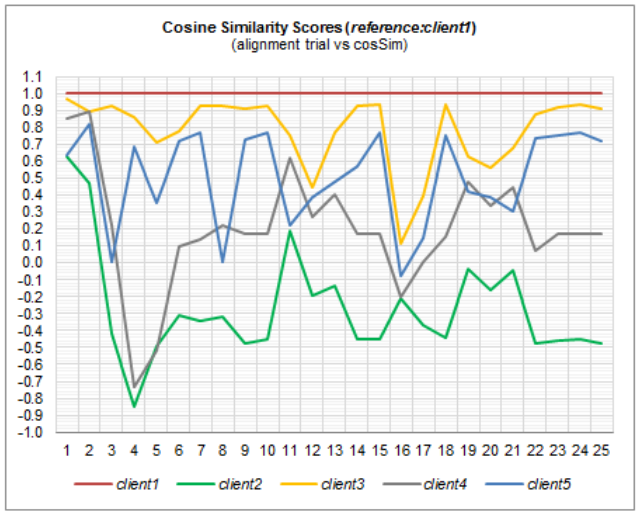


Figure E1.6. Cosine Similarity Score (cosSim) per Pairwise Alignment Run for Period End Closing Use Case (*reference:client1*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

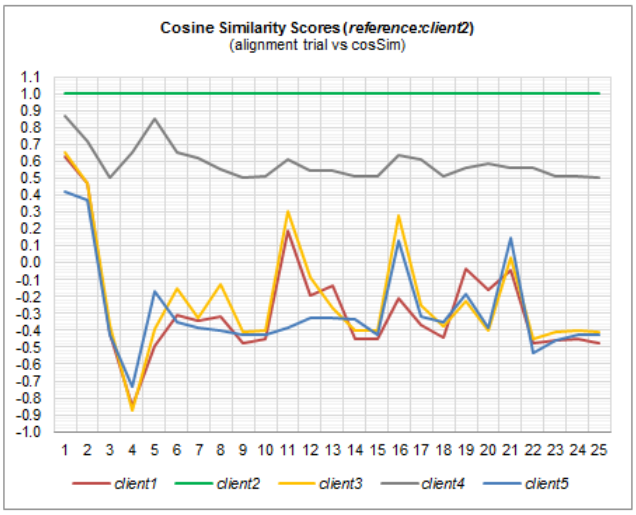


Figure E1.7. Cosine Similarity Score (cosSim) per Pairwise Alignment Run for Period End Closing Use Case (*reference:client2*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

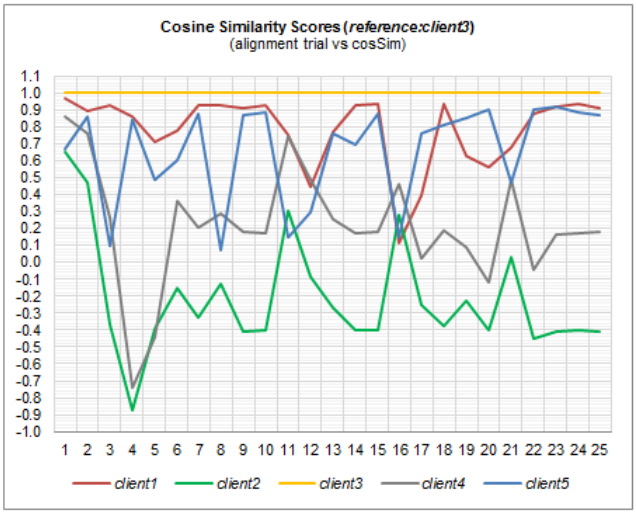


Figure E1.8. Cosine Similarity Score (cosSim) per Pairwise Alignment Run for Period End Closing Use Case (*reference:client3*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

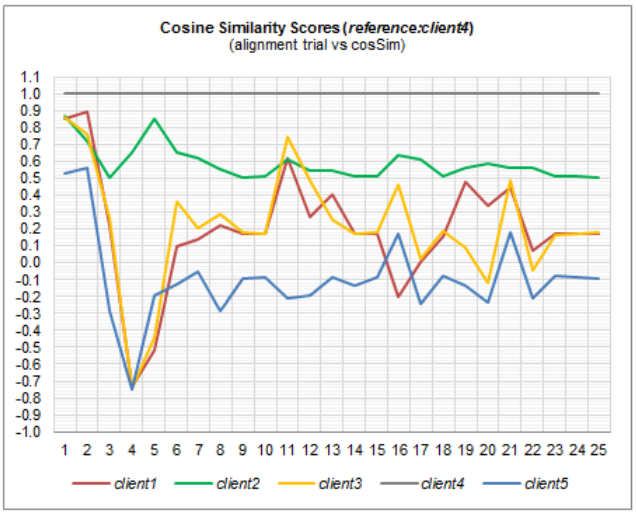


Figure E19. Cosine Similarity Score (cosSim) per Pairwise Alignment Run for Period End Closing Use Case (*reference:client4*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

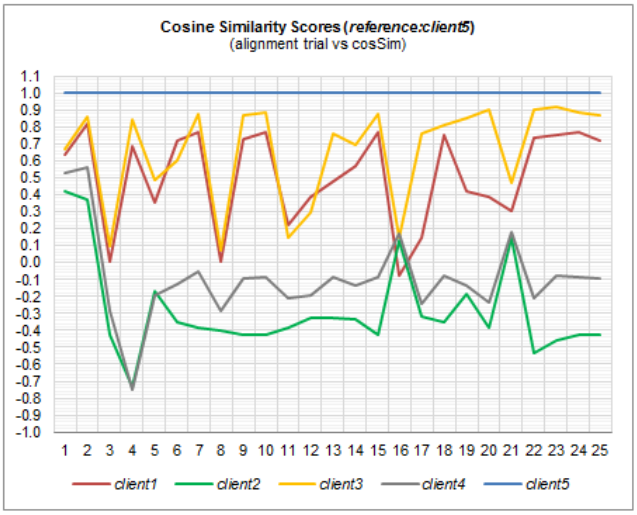


Figure E110. Cosine Similarity Score (cosSim) per Pairwise Alignment Run for Period End Closing Use Case (*reference:client5*) (*X-axis:alignment runID, Y-axis:cosine similarity score*).

Appendix E2

Appendix E2

APPENDIX F1 - Alignment Run List / Enviromental Permit Application

confEnhMSA Confidence-Enhanced Multi Sequence Alignment
Alignment Run List with Combined Individuals

RuntimeTable File: C:\academica\confEnhMSA_docu\wabo\input\waboruntime5.txt
ConfidenceTable File: C:\academica\confEnhMSA_docu\wabo\input\waboconfidence5.txt

AlignRunID: 1
Level: 4
Alignment: 50
Aligned source individual(s):

```
Seq3   A T J - B R - - - - - - - - - - - - - - - - - - - - - - - - O Z - - + - ? K
Seq5   A T - - B - - - R - G Z - - - - - - - - - - - - - - - - - - O J - - - - - - -
Seq1   A T - - B - - - - - G % J F D - - C - - - - - - - - - - R E O Q } - - P Z S
Seq2   A T - - - - - - - B - - - - - - - - - - - - - - - - J L M G C O U - R Z D F }
```

Aligned target individual(s):

```
Seq3   A T - J - - B R - - - - - - - - - - - - - - - - - - - - - O Z - - + - ? K
Seq5   A T - - - - B - - - - - - - - - R - - G Z - - - - - - - - - O J - - - - - -
Seq1   A T - - - - B - - - - - - - - - - - G % J F D C - - - R E O Q } - - P Z S
Seq2   A T - - - - - - - - - - - - - - - B - - - - - - - J L M G C O U - R Z D F }
```

simScr: 16.199335056356936
normSimScr: 1.0
strSimScr: 10.363536214954319
bhrSimScr: 5.835798841402617
alignment length: 37
optimal: false

Alignment: 51
Aligned source individual(s):

```
Seq3   A T J B R - - - - - - - - - - - - - - - - O Z - - + - ? - K -
Seq5   A T - B - R - G Z - - - - - - - - - - - - O J - - - - - - - -
Seq1   A T - B - - - G % J F D C - - - R E - - O Q } - - P Z - S -
Seq2   A T - - - - B - - - - - - J L M G C - - O U - R Z D F - } -
```

Aligned target individual(s):

```
Seq4   - A - - - - - - - - - - - - - - - G C F J O R - - - - - B T Z
```

simScr: 2.6685978769024885
normSimScr: 0.0
strSimScr: 1.0725130816766182
bhrSimScr: 1.5960847952258705
alignment length: 30
optimal: true

Alignment: 52
Aligned source individual(s):

```
Seq4   - A - - - - - - - - - - - - - - - G C F J O R - - - - - B T Z
```

Aligned target individual(s):

```
Seq3   A T J B R - - - - - - - - - - - - - - - - O Z - - + - ? - K -
Seq5   A T - B - R - G Z - - - - - - - - - - - - O J - - - - - - - -
Seq1   A T - B - - - G % J F D C - - - R E - - O Q } - - P Z - S -
Seq2   A T - - - - B - - - - - - J L M G C - - O U - R Z D F - } -
```

simScr: 2.6685978769024885
normSimScr: 0.0
strSimScr: 1.0725130816766182
bhrSimScr: 1.5960847952258705
alignment length: 30
optimal: false

Alignment: 53
Aligned source individual(s):
Seq4 A G C F J O R B T Z
Aligned target individual(s):
Seq4 A G C F J O R B T Z
simScr: 23.905579220844814
normSimScr: 1.0
strSimScr: 23.90557922084481
bhrSimScr: 0.0
alignment length: 10
optimal: false

Level: 3
Alignment: 41
Aligned source individual(s):
Seq3 A T J - B R - - - - - - - - - - - - - O Z - + - ? K
Seq5 A T - - B - - - R G Z - - - - - - - - O J - - - - -
Seq1 A T - - B - - - - G % J F D - - C R E O Q } - P Z S
Aligned target individual(s):
Seq3 A T - J - - B R - - - - - - - - - - - O Z - + - ? K
Seq5 A T - - - - B - R G Z - - - - - - - - O J - - - - -
Seq1 A T - - - - B - - G % J - - F D C R E O Q } - P Z S
simScr: 16.756188097987184
normSimScr: 1.0
strSimScr: 12.881188097987184
bhrSimScr: 3.875
alignment length: 26
optimal: false

Alignment: 42
Aligned source individual(s):
Seq3 A T J B R - - - - - - - - - - - - - O Z - - + - ? K
Seq5 A T - B - R - G Z - - - - - - - - - O J - - - - - -
Seq1 A T - B - - - G % J F D C - - - R E O Q } - - P Z S
Aligned target individual(s):
Seq2 A T - - - - B - - - - - - J L M G C O U - R Z D F }
simScr: 11.881427373422788
normSimScr: 0.6589024055759213
strSimScr: 6.909522637081311
bhrSimScr: 4.971904736341477
alignment length: 26
optimal: true

Alignment: 43
Aligned source individual(s):
Seq3 A T J B R - - - - - - - - - - O Z - + - - ? K
Seq5 A T - B - R G Z - - - - - - - O J - - - - - -
Seq1 A T - B - - G % J F D - C R E O Q } - - P Z S
Aligned target individual(s):
Seq4 - A - - - - - - - - - G C F J O R - - B T Z -
simScr: 2.464792309063151
normSimScr: 0.0
strSimScr: -0.5418228893623089
bhrSimScr: 3.00661519842546
alignment length: 23
optimal: false

APPENDIX F2 - Process Tree / Enviromental Permit Application

confEnhMSA Confidence-Enhanced Multi Sequence Alignment
Guide Tree List

Total Processing Time(sec): 164
RuntimeTable File: C:\academical\confEnhMSA_docu\wabo\input\waboruntime5.txt
ConfidenceTable File: C:\academical\confEnhMSA_docu\wabo\input\waboconfidence5.txt
Mode: Confidence Enhanced SA
MIN/MAX Normalization: true

AlignRunID: 1
Level: 5 (Root/Genesis)
Seq3 A T J B R - - - - - - - - - - - - - - - - O Z - - + - ? - K -
Seq5 A T - B - R - G Z - - - - - - - - - - - - O J - - - - - - - -
Seq1 A T - B - - - G % J F D C - - - R E - - O Q } - - P Z - S -
Seq2 A T - - - - B - - - - - - J L M G C - - O U - R Z D F - } -
Seq4 - A - - - - - - - - - - - - - - G C F J O R - - - - - B T Z
simScr: 2.6685978769024885
normSimScr: 0.0
strSimScr: 1.0725130816766182
bhrSimScr: 1.5960847952258705
individual length: 30

Level: 4
Seq3 A T J B R - - - - - - - - - - - - - O Z - - + - ? K
Seq5 A T - B - R - G Z - - - - - - - - - O J - - - - - -
Seq1 A T - B - - - G % J F D C - - - R E O Q } - - P Z S
Seq2 A T - - - - B - - - - - - J L M G C O U - R Z D F }
simScr: 11.881427373422788
normSimScr: 0.6589024055759213
strSimScr: 6.909522637081311
bhrSimScr: 4.971904736341477
individual length: 26

Seq4 A G C F J O R B T Z
individual length: 10

Level: 3
Seq3 A T J B R - - - - - - - - - O Z - + - ? K
Seq5 A T - B - R G Z - - - - - - O J - - - - -
Seq1 A T - B - - G % J F D C R E O Q } - P Z S
simScr: 12.866004649888554
normSimScr: 0.7304278420055201
strSimScr: 11.243581852250369
bhrSimScr: 1.622422797638185
individual length: 21

Seq2 A T B J L M G C O U R Z D F }
individual length: 15

Seq4 A G C F J O R B T Z
individual length: 10

Level: 2
Seq5 A T B R G Z - - - - - - O J - - - -
Seq1 A T B - G % J F D C R E O Q } P Z S
simScr: 13.21937797858329
normSimScr: 0.9358632835942571
strSimScr: 11.269377978583288
bhrSimScr: 1.9500000000000028
individual length: 18

Seq2 A T B J L M G C O U R Z D F }
individual length: 15

Seq3 A T J B R O Z + ? K
individual length: 10

Seq4 A G C F J O R B T Z
individual length: 10

Level: 1
Seq1 A T B G % J F D C R E O Q } P Z S
individual length: 17

Seq2 A T B J L M G C O U R Z D F }
individual length: 15

Seq3 A T J B R O Z + ? K
individual length: 10

Seq4 A G C F J O R B T Z
individual length: 10

Seq5 A T B R G Z O J
individual length: 8

CURRICULUM VITAE

Personal Information

Eren Esgin was born in Bandırma on February 8, 1982. He received his B.Sc. degree in Industrial Engineering from Middle East Technical University in 2004 and M.Sc. degree in Information Systems from Informatics Institute at Middle East Technical University in 2009. He has been working as a SAP consultant for more than 12 at SAP EPM (Enterprise Performance Management) and CO/PS (Controlling and Project Systems) modules.

Education

2009 – present, Ph.D., Information Systems at Informatics Institute, Middle East Technical University (METU), Ankara Turkey
2006 – 2009, M.Sc., Information Systems at Informatics Institute, Middle East Technical University (METU), Ankara Turkey Thesis Title: *A Hybrid Methodology in Process Modeling: From-to Chart Based Process Discovery*
2000 – 2004, B.Sc., Industrial Engineering Department, Middle East Technical University (METU), Ankara Turkey

Work Experience

Organization	Duration	Position
Vektora	May 2018 – present	SAP Solutions Architect
ACRON	March 2017 – April 2018	Senior SAP EPM Consultant
Turing Analytics	October 2015 – February 2017	Founder Freelance SAP Consultant
Metric	February 2015–September 2015	SAP Solutions Architect
ACRON	January 2014 – January 2015	SAP EPM Consultant
MBIS	February 2012 – December 2013	SAP BPC/CO/PS Consultant
HAVELSAN	June 2004 – August 2011	SAP Financial Applications Consultant

Publications

Esgin, E. & Karagoz, P. (2015) *Dynamic Scoring-based Sequence Alignment for Process Diagnostics*. Current Approaches in Applied Artificial Intelligence, vol. 9101, 742–752.

Esgin, E. & Karagoz, P. (2013) *Confidence-Aware Sequence Alignment for Process Diagnostics*. International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 990–997.

Esgin, E. & Karagoz, P. (2013) *Sequence Alignment Adaptation for Process Diagnostics and Delta Analysis*. 8th International Conference, Hybrid Artificial Intelligence Systems (HAIS), vol. 8073, 191–201.

Esgin, E. & Senkul, P. (2011) *Delta Analysis: A Hybrid Quantitative Approach for Measuring Discrepancies between Business Process Models*. 6th International Conference, Hybrid Artificial Intelligence Systems (HAIS), vol. 6678, 296–304.

Esgin, E. & Senkul, P. (2011) *Extracting Connection Types in Process Models Discovered by Using From-to Chart Based Approach*. Developing Concepts in Applied Intelligence, vol. 363, 59–65.

Esgin, E., Senkul, P. & Cimenbicer, C. (2010) *A Hybrid Approach for Process Mining: Using From-to Chart Arranged by Genetic Algorithms*. 5th International Conference, Hybrid Artificial Intelligence Systems (HAIS), vol. 6076, 178–186.

Esgin, E. & Senkul, P. (2009) Hybrid Approach to Process Mining: Finding Immediate Successors of a Process by Using From-to Chart. International Conference on Machine Learning and Applications (ICMLA), 664–668.

Awards

Vestel Customer Services Spare Parts Prediction on SAP Predictive Analytics Project awarded as "*Best Data Mining Project of 2016"* by SAP Turkey.