A comparative study of existing quality measures for process discovery

Peer-reviewed author version

JANSSENSWILLEN, Gert; Donders, Niels; JOUCK, Toon & DEPAIRE, Benoit (2017)
A comparative study of existing quality measures for process discovery. In:
INFORMATION SYSTEMS, 71, p. 1-15.

DOI: 10.1016/j.is.2017.06.002
Handle: http://hdl.handle.net/1942/25149

A comparative study of existing quality measures for process discovery

Gert Janssenswillen[a,b,*], Niels Donders[a], Toon Jouck[a], Benoît Depaire[a]

[a] *UHasselt - Hasselt University, Faculty of Business Economics, Agoralaan, 3590 Diepenbeek, Belgium*
[b] *FWO - Flemish Research Foundation, Egmontstraat 5, 1000 Brussels, Belgium*

Abstract

Evaluating the quality of discovered process models is an important task in many process mining analyses. Currently, several metrics measuring the fitness, precision and generalization of a discovered model are implemented. However, there is little empirical evidence how these metrics relate to each other, both within and across these different quality dimensions. In order to better understand these relationships, a large-scale comparative experiment was conducted. The statistical analysis of the results shows that, although fitness and precision metrics behave very similar within their dimension, some are more pessimistic while others are more optimistic. Furthermore, it was found that there is no agreement between generalization metrics. The results of the study can be used to inform decisions on which quality metrics to use in practice. Moreover, they highlight issues which give rise to new directions for future research in the area of quality measurement.

Keywords: Process Discovery, Process Quality, Conformance Checking, Process Metric

1. Introduction

In recent times, organizations possess a tremendous amount of data concerning their customers and products. Many activities which take place in their operational processes are being recorded in event logs [28]. Techniques from the process mining field, which has grown steadily over the last decades, can be applied to gain insights in these event data [27]. In recent years, a lot of attention has been given to the discovery of process models from event logs [11, 18, 29, 35], and subsequently, the quality measurement of these models

*Corresponding author
Email addresses: gert.janssenswillen@uhasselt.be (Gert Janssenswillen), ndonders93@gmail.com (Niels Donders), toon.jouck@uhasselt.be (Toon Jouck), benoit.depaire@uhasselt.be (Benoît Depaire)
URL: www.businessinformatics.be (Gert Janssenswillen)

[1, 2, 6, 25, 26]. Assessing the quality of discovered models is essential in order to find out whether it constitutes an appropriate representation of the process. The quality of discovered process models has been broken down in four dimensions: fitness, precision, generalization and simplicity [22]. For each of the dimensions, several metrics have been implemented, an overview of which can be found in vanden Broucke et al. [32].

Although the existing metrics have been used to compare the performance of process discovery algorithms [9], little research has been done concerning the evaluation and comparison of the metrics itself. Until now, it is unclear what the differences are between metrics within the same dimension: do they judge discovered process models in a similar way, or do they qualify models differently? Are some metrics more optimistic or pessimistic than others? Furthermore, there is ongoing debate about the precise definition of certain dimensions, and the relationships between the dimensions. Nevertheless, it is essential to know which quality dimensions to take into account given a specific use case and which measures are most suitable to be used.

In this paper, we conduct an empirical study, incorporating the state-of-the-art quality metrics, with the aim to statistically analyze the relationships between metrics within and among dimensions. The results of the experiments indicate:

- the feasibility of the metrics, in terms of CPU-time and memory,
- whether metrics measuring the same dimension agree with each other or not,
- whether the dimensions are related to each other, or independent from one another,
- to which extent some metrics are more optimistic about process model quality compared to others,
- to which extent some metrics are more sensitive to differences in process models quality compared to others.

The next section further introduces the different dimensions and the related metrics which are subject of the analysis. Section 3 discusses the experimental set up. The results of the experiment are reported and discussed in Section 4. Section 5 concludes the paper.

2. Related work

In this section, the quality dimensions are further introduced and discussed. Subsequently, an overview is given of different metrics that have been implemented. Finally, related empirical work is discussed.

Table 1: Quality Dimensions of Discovered Process Models.

Dimension	**Description**
Fitness	A model with good fitness allows for the behavior seen in the event log
Precision	A model is precise if it does not allow for too much observed behavior
Generalization	A model should generalize and not restrict behavior to the examples seen in the event log
Simplicity	A model should be as simple as possible, and easy to understand.

2.1. Quality Dimensions

The quality of a discovered process model has been broken down into four different quality dimensions [22], as displayed in Table 1. Firstly, the fitness dimension measures the extent to which the discovered process model is able to replay the behavior seen in the event log.

Secondly, the precision dimension states that the model should be precise and not contain behavior which was not observed. When both fitness and precision are optimized, the model contains all the recorded behavior and nothing more.

Thirdly, generalization specifies that models should generalize and not only restrict behavior to the sample contained by the event log. In other words, a model with high quality should also be able to replay previously unseen behavior from the process.

Finally, it is said that, according to the Simplicity dimension, simpler models are preferred over complex ones. On the definition of *simpler* models, two different interpretations exist. On the one hand, simpler models have been defined as models which are not overly complex, e.g., they are not extremely large and the density of arcs is low [19]. On the other hand, some have defined simplicity of models as understandability, which places more emphasis on the ease of interpretation and cognitive capabilities [24].

It should be noted that an inherent trade-off between the dimensions of precision and generalization exists, as a model can not generalize to unseen behavior and be precise at one and the same time. Recent literature proposes a new evaluation framework, in which the quality dimensions to be evaluated depend on the objective of the quality measurement [16]. When one wants to quantify whether the model is a good representation of the behavior in the event log, the dimensions log-fitness and log-precision are proposed. Both dimensions are defined in the same way as the traditional fitness and precision dimensions, but their new name emphasizes that they measure the quality of a model with respect to the log.

However, when the objective is to quantify whether the discovered model is a good representation of the underlying process, i.e. the system, the dimensions system-fitness and system-precision are suggested. The latter two dimensions

are thus suggested to replace the generalization dimension. The term *system* refers to the real, underlying process. It defines the actual way work can be done and it is generally unknown. The event log is regarded as a sample of the system behavior.

By making the dimensions to quantify contingent on the objective of the measurement, trade-offs between dimensions are removed and explicitly translated into a trade-off between objectives. The proposed framework in Janssenswillen et al. [16] accommodates for the fact that different ambiguous and contradicting definitions exist for generalization. The empirical results indicated that the generalization metrics were negatively correlated, suggesting that they are not measuring a singular aspect of model quality. As a result, a comparative study of the behavior of existing metrics is imperative.

2.2. Quality metrics

In this paper, the behavior of state-of-the-art quality metrics is analyzed using a varying set of models and event logs, starting from the original quality paradigm. As the focus of this paper lies exclusively on the fit between model and log, only the first three dimensions are discussed, i.e. fitness, precision and generalization. Indeed, simplicity focuses only on the model, and does not relate to the event log. Making a model simpler is sometimes also view as a preprocessing-step which can be performed after the other quality criteria are checked on [10]. An overview of the existing metrics can be found in Table 2, which is based on the overview given in vanden Broucke et al. [32]. Most research has been attributed to fitness and precision metrics, while only a limited amount of work is available on generalization.

2.2.1. Fitness

Fitness was one of the first quality dimensions for which metrics were implemented. The first metrics were rather coarse-grained and naive and directed to a specific set of models. Afterwards more advanced, fine-grained metrics for Petri Nets have become available. An overview of the metrics in given in Table 2 and more detailed descriptions are provided below.

- The **Parsing Measure** metric [35] is defined as the percentage of correctly parsed traces in the event log, and is therefore a quite coarse-grained metric.
- The **Continuous Parsing Method** [35] is slightly more fine-grained as it records errors and then continues parsing. As such, it is defined as the percentage of successfully parsed *events*. As well as the Parsing Measure, it expects a heuristics net as input.
- The **Completeness** metric as described in Greco et al. [13], is defined in the same way as the Parsing Measure, with the only difference that it expects a workflow schema as input. Consequently, Completeness is also a coarse-grained, naive metric.

Table 2: Overview of Existing Quality Metrics for Fitness (F), Precision (P) and Generalization (G).

	Metric	**Author**	**Date**	**Range**	**Model Input type**	**Included**
F	Parsing Measure	Weijters et al. [35]	2006	$[0, 1]$	Heuristics Net	
	Continuous Parsing method	Weijters et al. [35]	2006	$[0, 1]$	Heuristics Net	
	Completeness	Greco et al. [13]	2006	$[0, 1]$	Workflow Schema	
	Partial Fitness - complete	Alves de Medeiros [7]	2007	$[-\infty, 1]$	Heuristics Net	
	Token Based Fitness	Rozinat et al. [23]	2008	$[0, 1]$	Petri Net	•
	Proper Completion	Rozinat et al. [23]	2008	$[0, 1]$	Petri Net	
	Behavioral Recall	Goedertier et al. [12]	2009	$[0, 1]$	Petri Net	•
	Behavioral Profile Conformance	Weidlich et al. [34]	2011	$[0, 1]$	Petri Net	
	Alignment Based Fitness	van der Aalst et al. [26]	2012	$[0, 1]$	Petri Net	•
P	Soundness	Greco et al. [13]	2006	$[0, 1]$	Workflow Schema	
	(Advanced) Behavioral Appropriateness	Rozinat et al. [23]	2008	$[0, 1]$	Petri Net	
	Behavioral Specificity	Goedertier et al. [12]	2009	$[0, 1]$	Petri Net	
	ETC-Precision	Munoz-Gama et al. [20]	2010	$[0, 1]$	Petri Net	
	Alignment Based Precision	van der Aalst et al. [26]	2012	$[0, 1]$	Petri Net	•
	(weighted) Behavioral Precision	vanden Broucke et al. [31]	2014	$[0, 1]$	Petri Net	•
	One Align Precision	Adriansyah et al. [2]	2015	$[0, 1]$	Petri Net	•
	Best Align Precision	Adriansyah et al. [2]	2015	$[0, 1]$	Petri Net	•
G	Alignment Based Generalization	van der Aalst et al. [26]	2012	$[0, 1]$	Petri Net	•
	Frequency of use	Buijs et al. [3]	2014	$[0, 1]$	Process Tree	
	(weighted) Behavioral Generalization	vanden Broucke et al. [31]	2014	$[0, 1]$	Petri Net	•

- The **Partial Fitness - Complete** metric, originally defined in de Medeiros [7], is similar to the Continuous Parsing Method, to the extent that it expects a heuristics net and it is a fine-grained metric. However, it does not only count activities which can be parsed but also punishes for tokens which are left behind. As a result, the range of possible values for this metrics extends from $-\infty$ to 1.

- The **Token Based fitness** metric [23] is one of the first fitness metrics for Petri Nets, and is specifically based on their execution semantics. In order to quantify fitness, the event log is replayed on a Petri Net representation of the discovered model, during which penalties are given when tokens are missing to execute the next transition. Likewise, penalties are allocated for tokens which remain in the model after replaying. Despite the straightforwardness of this metric, it has a few disadvantages. Firstly, the reliance on tokens creates a strong representational bias: two Petri Nets which are trace equivalent but have a different composition of places and transitions can have different values for this metric. Furthermore, due to state space explosion, the calculation of the values might be problematic, especially in the presence of silent transitions.

- The **Proper Completion** metric [23] is the Petri-net based alternative to the Parsing Measure and Completeness metric, as it is defined as the percentage of traces without any missing of remaining tokens after trace replay. It can thus be regarded as a coarse-grained, naive version of Token Based Fitness.

- The **Behavioral Recall** metric [12] relies on a technique which induces so-called *negative events*. These are events which are supposed to be not allowed at a certain point in the process. Inducing negative events requires the configuration of considerably more parameters compared to the other metrics, which require a higher amount of background knowledge about the algorithm. This also make the metric more sensitive, as there are many more factors to take into account.

- The **Behavioral Profile Conformance** Metrics defined in Weidlich et al. [34] are a set of metrics which relate to different constraints imposed by a model, such as precedence relations and co-occurence of activities. It is therefore fundamentally different as the other metrics quantify fitness with a single value.

- The **Alignment-Based Fitness** metric [26] compares sequences of activities. Each sequence of activities in the log is aligned to an execution sequence in the model based on a certain cost-configuration for insertions and deletions. The fitness for a single case is then determined based on the cost of the optimal alignment for that case, while the overall fitness of an event log with respect to a model, is the average of the fitness values for all the cases. Performance issues might exist, as the optimal alignment

for each trace in the event log needs to be found, which is computationally expensive.

2.2.2. Precision

While a few early precision metrics exist, most research on the precision dimension originated later compared to fitness. Recently, new metrics have been developed which combine existing approach or introduce new ones. An overview of the metrics in given in Table 2 and more detailed descriptions are provided below.

- The **Soundness** metric [13] can be regarded as the precision counterpart of the Completeness fitness metric. It is defined as the number of traces in a model which is also part of the model. As for Completeness, a workflow schema is expected as input.
- The **(Advanced) Behavioral Appropriateness** metric [23] is a footprint-based metric which compares *follows* and *precedes* relationships. It is rather coarse-grained and computationally expensive, as it requires a state space exploration.
- The **Behavioral Specificity** metric [12] uses the induction of negative events. It is defined as the percentage of correctly classified negative events, i.e. events that should not be able to happen because they were regarded as negative, and which are indeed not allowed in the model.
- The **ETC Precision** metric [20] uses the notion of escaping arcs in a prefix automaton to measure precision. However, this approach cannot cope with non-fitting event logs.
- The **Alignment-Based Precision** metric [26] builds on the same concepts as the Alignment Based Fitness, as it is calculated based on an *aligned* event log. This means that each non-fitting trace is replaced with the execution trace of the model to which it was aligned. Then, for each state, the metric compares the number of different activities that have occurred to the total number of activities possible in the model. When this ratio is low, it suggests that the model is imprecise.
- The **(weighted) Behavioral Precision** metric [31] is also based on the induction of negative events. Here, precision is defined as the ratio between the number of *True Positive* events - observed events which can be replayed - on the one hand, and all positive events on the other hand - all events which can be replayed by the model, both observed and not observed - on the other hand. When only observed events can be replayed by the model, the model is precise. Note that it differs from Behavioral Specificity in that the latter metric takes into account which events are negative, while Behavioral Precision does not take this into account.

- The **One Align Precision** metric [2] is based on the ETC-precision metric [20]. While ETC precision cannot be used for non-fitting event logs, One Align allows the computation of the ETC-precision metric by first aligning the event log with the model. The name One Align stems from the fact that, when multiple optimal alignments exist, only one is taken into account

- The **Best Align Precision** metric [2] is similar to the One Align metric, but it takes into account all the optimal alignments which exist.

2.2.3. Generalization

Indisputably, the generalization dimension has received the least attention. Nevertheless, a few metrics have been developed, some of which very recently. An overview of the metrics in given in Table 2 and more detailed descriptions are provided below.

- The **Alignment Based Probabilistic** metric [26] is related to the work on Alignment Based Fitness and Alignment Based Precision, and attempts to estimate the probability that a new unobserved case can be replayed by the current model, using bayesian statistics.

- The **Frequency of use** metric [3] is a generalization metric defined for process trees which estimates the generalization by looking at the frequencies of executions in the process tree. When certain parts of the process tree are infrequent, the tree is regarded as overfitting, and thus has a lower generalization.

- The **Behavioral Generalization** metric [31] was introduced in the literature related to negative events. In this case, the presence of negative events can be used to measure generalization, as it provides a distinction about which events the model should be able to replay, and which it should not. In particular, generalization is defined here as the ratio between the number of *allowed generalizations* on the one hand, and the total number of generalizations - on the other hand. The set of generalizations is defined as the events which are nor observed nor classified as negative. Allowed generalizations are the subset of generalizations which are allowed be the model. The more generalizations are allowed, the higher the generalization metric will be.

2.3. Related empirical work

Literature on evaluating and comparing quality metrics is limited, although some works should be noticed. In Rozinat et al. [22], metrics were compared at a very small scale. However, as this is one of the earliest works on process model quality, most of those metrics have become obsolete. The metrics based on negative events were incorporated in a comparison in De Weerdt et al. [8], but also here only a small set of example models was used. Nevertheless the

authors concluded that not all metrics are one-dimensional and some suffer from computational inefficiency.

Experiments on a much larger scale were done in De Weerdt et al. [9], although the objective of this research was to compare the performance of discovery algorithms. Therefore, no conclusions on the relationship between metrics within and among dimensions were drawn. Finally, in vanden Broucke et al. [30], metrics were compared within dimensions. Here, the hypothesis that the average of different metrics within each dimension were equal was rejected. Nonetheless, no further analyses on their relationship were done.

Compared with the existing literature, the contribution of this paper is that the state-of-the-art quality metrics are evaluated on a large set of event logs and models. The focus is not to compare discovery algorithms, but rather to compare the measurements of the quality metric itself. The gained insights can then be used to make an informed decision on which quality metrics to use for the evaluation of discovered process models.

3. Methodology

The methodology used in this paper is based on the framework for comparing process mining algorithms presented in Weber et al. [33]. In particular, the experiment encompasses the steps listed below. Each of these is discussed in more detail in the remainder of this section. The summary of the experiment can be found in Table 3 and a schematic overview is given in Figure 1. [1]

Step 1. Generation of systems
Step 2. Determine of number of execution paths
Step 3. Generation of logs
Step 4. Model Discovery
Step 5. Quality Measurement
Step 6. Empirical Analysis

3.1. Generation of systems

As a first steps, systems are generated which are to act as ground truth process models. The systems were generated using the methodology described in Jouck and Depaire [17]. As input for this generation, different population parameters had to be set, such as the distribution for the number of leaf nodes, the distribution for the type of operator nodes and the probability for silent and duplicate tasks. Table 4 shows the used population parameters for each of the 15 systems. For more information on how these parameters are used to generate the process tree, we refer to Jouck and Depaire [17].

[1] All the systems, logs, models and quality measurements can be found online and are available to be used for other experiments: `https://github.com/gertjanssenswillen/processquality/`.

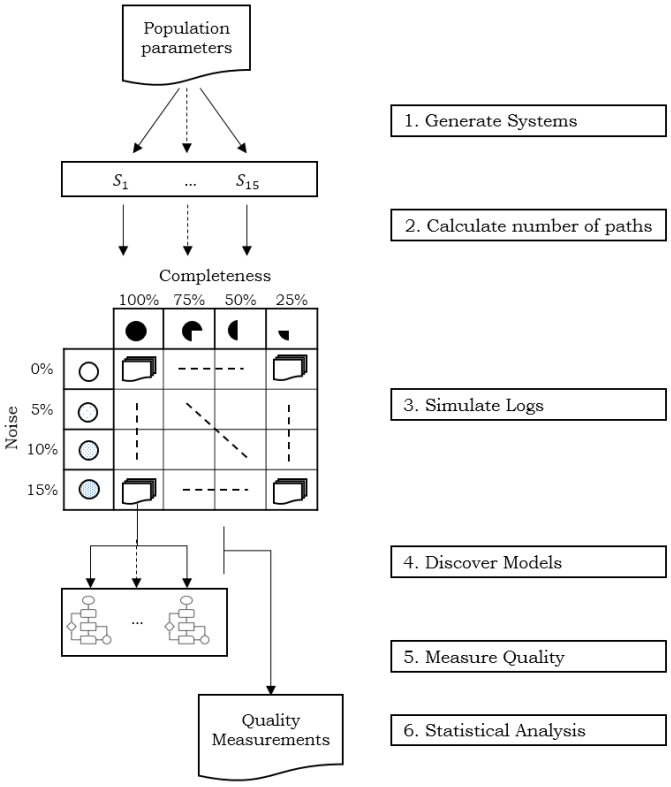


Figure 1: Schematic overview of experimental setup.

Table 3: Experimental setup.

Step	Characteristic	Value
1	Number of systems	15
3	Completeness Levels	100%, 75%, 50%, 25%
	Noise levels	0%, 5%, 10%, 15%
	Number of logs	1200 logs
4	Discovery algorithms	Heuristics[35]
		Inductive[18]
		ILP [29]
		Alpha Miner [28]
		Flower Miner
	Number of models	6000 models
5	Fitness	Token Based Fitness [23]
		Behavioral Recall [12]
		Alignment Based Fitness [26]
	Precision	Alignment Based Precision [26]
		Behavioral Precision [31]
		One Align Precision [2]
		Best Align Precision [2]
	Generalization	Alignment Based Generalization [26]
		Behavioral Generalization [31]

In terms of parameters, two groups of systems were generated: systems of moderate complexity (MP_1-MP_{10}) and systems of high complexity (MP_{11}-MP_{15}). This is inspired by the findings in De Weerdt et al. [9], where it was found that process discovery algorithms perform differently when the process behavior is complex (real life event logs) instead of more elementary process behavior (artificial event logs). As such, the obtained values for the quality metrics are expected to be more widespread over the range from zero to one. Complexity in this context is related to the mix of constructs which are used in the systems as well the number of leave nodes. Systems with a higher complexity have more leave nodes, and thus activities, and have a higher proportion of more advanced constructs, such as loops and inclusive choice. Moreover, the probability of long-term dependencies and duplicate tasks is higher. For example, MP_1 to MP_{10} contain models with on average 15 visible activities, while the other populations contain models with on average 20 visible activities. Furthermore, only one model population among the first group has long-term dependencies (i.e. MP_9), while four out of five of the more complex populations

have this property.

Note that only a single system was generated from each of the model population. Therefore, it is not possible to draw any conclusions about the impact of the parameters, and thus the type of models, on the experiment. Relating the behavior of process quality metrics to characteristics of the process is out of the scope of this paper. Rather, the population parameters were set in this way to include a wide variety of process models in the analysis. As a result, the selection of a single system for each model population does not hamper the results of this study.

3.2. Determine the number of execution paths

In order to further increase the variability in the event data, and thereby bringing about the discovery of a large set of different models, logs with a different level of completeness and noise are generated in the next step. To be able to target the completeness of event logs, the number of execution paths in each of the systems needs to be calculated first. The algorithm introduced in Janssenswillen et al. [15] was used. In this calculation, loops were only allowed to be iterated over a maximum number of three times, to avoid an infinite number of paths. While this appears to be restrictive, the number of paths is only used as a reference point to create logs with a differing level of completeness. Both completeness and noise as a characteristic of the event log will not be used explicitly during the analysis.

3.3. Generation of logs

For each of the systems, event logs with a certain level of completeness and noise have been simulated. For completeness, 4 levels were considered: 100%, 75%, 50%, and 25%. These percentages measure how many of the different paths in the system, as calculated in the previous step, have been observed in the event log. Thus, for a model with 100 unique paths, a log with 75% completeness is one where 75 of the unique paths in the system have been seen.

Analogously, 4 different noise levels were considered: 0%, 5%, 10%, 15%. A log with 15% of noise means that 15% of the cases contain noise. The types of noise that where induced are described in Jouck and Depaire [17]. For each of the systems (15) and each of the noise (4) and completeness (4) levels, 5 different logs were generated. This amounted to a total of $15 \cdot 4 \cdot 4 \cdot 5 = 1200$ logs.

3.4. Model discovery

Afterwards, the simulated logs were used for the discovery of process models. For each log, five different process discovery algorithms were applied: the Alpha Miner [28], the Heuristics Miner [35], the Inductive Miner [18], the ILP miner [29] and the Flower Miner. Note that the goal of the experiment is not to evaluate the performance of these miners. However, a variety of mining algorithms has been selected in order to avoid algorithm-specific biases. The main goal of the process discovery step is thus to provide a large variety of models

Table 4: Population parameters.

Parameters	Population														
	MP_1	MP_2	MP_3	MP_4	MP_5	MP_6	MP_7	MP_8	MP_9	MP_{10}	MP_{11}	MP_{12}	MP_{13}	MP_{14}	MP_{15}
Minimum of visible activities	10	10	10	10	10	10	10	10	10	10	15	15	15	15	15
Mode of visible activities	15	15	15	15	15	15	15	15	15	15	20	20	20	20	20
Maximum of visible activities	20	20	20	20	20	20	20	20	20	20	25	25	25	25	25
Sequence ($\Pi^{\rightarrow}$)	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.45	0.45	0.35	0.30	0.45	0.45
Parallel ($\Pi^{\wedge}$)	0.30	0.00	0.15	0.15	0.00	0.10	0.10	0.10	0.10	0.00	0.10	0.05	0.00	0.10	0.00
Choice ($\Pi^{\times}$)	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.40	0.20	0.20	0.30	0.20	0.20
Loops ($\Pi^{\circlearrowleft}$)	0.00	0.30	0.15	0.00	0.15	0.10	0.10	0.10	0.10	0.00	0.25	0.30	0.30	0.25	0.25
Or ($\Pi^{\vee}$)	0.00	0.00	0.00	0.15	0.15	0.10	0.10	0.10	0.10	0.15	0.00	0.10	0.10	0.00	0.10
Silent activities(Π^{τ})	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.10	0.10	0.00	0.00	0.00
Reoccuring activities (Π^{Re})	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.10	0.00	0.10	0.00	0.10
Long-term dependencies (Π^{Lt})	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.50	1.00	0.00	0.50	1.00

of which the quality can be measured by different metrics. Each of the algorithms returned a Petri Net, of which the quality is measured in the next step. ProM 6.5 was used for the discovery of the process models. Default values were used for all parameters. In total, 1200 logs $\cdot$ 5 algorithms = 6000 models were discovered.

3.5. Quality Measurement

The metrics which were used for quality measurement in this experiment are indicated in the last column of Table 2. The metrics were selected first and foremost based on their expected type of model input, as all discovery algorithms used return a Petri Net. Furthermore, the initial coarse-grained metrics such as Proper Completion and (Advanced) Behavioral Appropriateness are not included. While Behavioral Precision was included, Behavioral Specificity was not considered, as it is defined slightly different compared to other precision metrics, as stated in vanden Broucke et al. [32]. Since the discovery algorithms used do not guarantee a perfect fitness, also ETC-precision is not taken into account.

Each of the metrics was calculated for each model against the event log it was discovered from. The resulting values are the input for the experimental analysis. All calculations were performed using the benchmarking framework CoBeFra [32]. In total, 6000 models $\cdot$9 metrics = 54000 metrics were computed.

3.6. Empirical Analysis

The obtained values are thereafter statistically analyzed. In particular, the metrics are investigated on three desirable properties: feasibility, validity and sensitivity.

3.6.1. Feasibility

One should be able to assess the quality of a model within a reasonable amount of time and without excessive memory capacity. In order to test this, the calculations are performed with a limited, though not unreasonable amount of resources. In particular, a maximum working memory of 1Gb is used and computations are not allowed to last more than one hour.

3.6.2. Validity

The validity of the metrics is assessed, i.e. whether they measure what they are supposed to measure. In order to do this, the relationships between metrics within and among dimensions are analyzed by means of a correlation analysis and a factor analysis.

The analysis of correlations reveals whether metrics within a specific dimension are positively correlated with each other or not. Furthermore, by examining the correlations across different dimensions, the relations between the dimensions will become clear.

Secondly, an Exploratory Factor Analysis (EFA) [14] is conducted. Since the set of dimensions is not unanimously accepted in literature, an Exploratory

Factor Analysis (EFA) is chosen instead of a Confirmatory Factor Analysis. This allows non a priori specified factors to be found. In order to decide on the number of factors to construct, a scree plot is be composed to find the number of factors that explain the most variability in the data. In order to make the factors more interpretable, a rotation has been applied. A Promax rotation is chosen [5]. This is an oblique, non-orthogonal rotation, which assumes that factors are possibly correlated. Since it is not clear whether dimensions (or their implementations) are orthogonal or not, an oblique rotation is the safest option.

3.6.3. Sensitivity

Finally, the sensitivity of the metrics is investigated. Both the analysis of factors and correlations implicitly assume that the relations between different metrics are the same for the complete range of values, i.e. they behave the same for models with good quality as well as for models with bad quality. Nonetheless, it is not impossible that metrics agree on the precision of very precise models, while they judge the precision of less precise models differently. By comparing all metrics pairwise, it becomes clear whether some metrics are more pessimistic than others. Furthermore, it clarifies whether certain metrics observe differences between models where others do not, and thus are more sensitive.

For each pair of metrics within a dimension, the relationship is analyzed by drawing a scatter plot and fitting a Lowess smoothing line onto it [4]. This smoothing line can then be compared to the diagonal. For example, when the smoothing line approximates the diagonal, the two metrics at hand score models equally. However, when the smoothing line falls below the diagonal, the y-axis metric is more pessimistic. When it sits above the diagonal, the y-axis metric is more optimistic. Moreover, when the *slope* of the Lowess curve significantly differs from the diagonal, it can be said that there is a difference in sensitivity. I.e. when the Lowess curves turns toward a specific metric, it can be said that this metric becomes less sensitive to differences in quality compared to the other metric.

4. Results

The median log size of the generated event logs was 10,704 events, with an overall minimum of 247 and a maximum of 530,203. Each log contained 3,649 cases on average, while the average number of distinct activity sequences was 545. It should be noted that for the logs generated from systems with a moderate complexity, the median log size was only 5,670 events, while for logs with a high complexity, this was 13,904 events.

4.1. Feasibility

During the computation of the metrics, it turned out that some of the computations could not be completed because of excessive requirements in memory or CPU time. For each computation 1Gb of working memory was available and computations were aborted after 1 hour. Figure 2 gives an overview of the

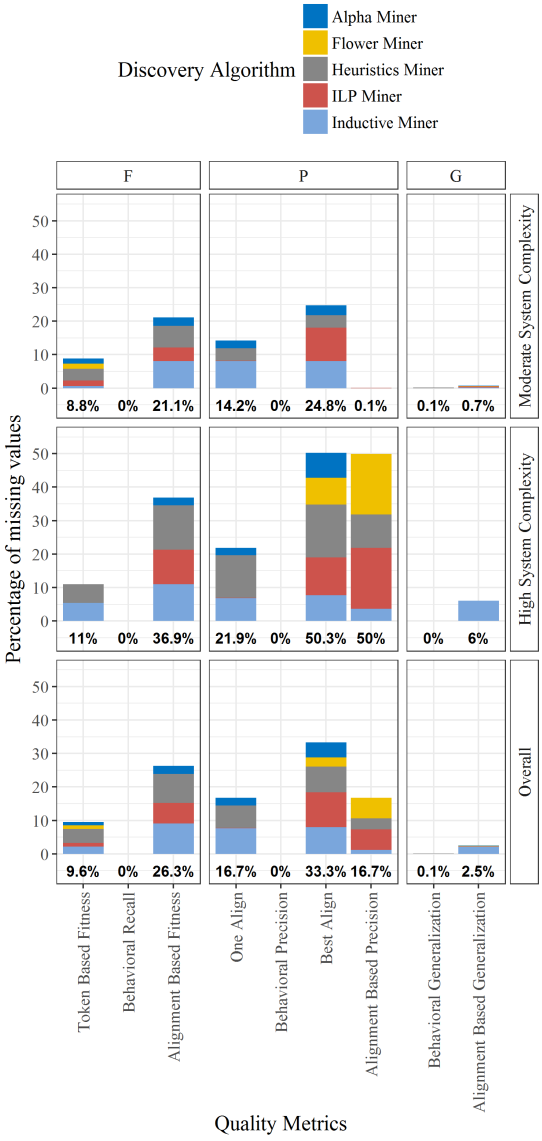


Figure 2: Missing values by metric and system complexity.

computations that were not complete. This shows that precision metrics suffered more from this problem, in particular the Alignment Based metrics. By comparing the number of missing values conditioned on the complexity of the system from which the logs were simulated, as was defined earlier, it is clear that more problems occurred for the systems with a higher complexity. For Alignment Based Precision and Best Align Precision, the percentage of missing

values increases to 50% for systems with high complexity.

It should be noted here that these problems are not only caused by (the implementation of) the metrics itself, but also by the quality of the models which are discovered. The colors in Figure 2 show the distribution of the missing values among the different miners. It is clear the the missing values are not spread randomly among the miners, but instead, some of the miners create models for which quality measurement by some of the metrics gets unfeasible. For example, the problems with Alignment Based Precision and Best Align Precision are mainly related to models discovered by the Flower Miner, ILP miner and Heuristics Miner. This can be explained because these algorithms tend to discover models which allow for too much behavior (cfr. Figure 5). As a result, it is computationally hard to find the optimal alignment between the log and the model. On the other hand, the Behavioral Precision metric has no problem with finding a value for these models, while One Align Precision mainly has a problem with models from the Heuristics Miner.

It can thus be concluded that, when the complexity of the behavior is high, some of the metrics are not suitable to use in practice in combination with certain discovery algorithms. In particular, for models which contain a large number of different activity execution sequences, metrics which rely on alignments experience difficulties to measure the precision.

Overall, the percentage of missing values was 11.69%. For 2952 (49.2%) models all values were obtained, i.e. for all 9 metrics. Only these *complete* observations are used in the remainder of the analysis.Since the missing values are related to specific type of models (i.e. imprecise) models, it would be unfair to use partial observations in the analysis.

4.2. Validity

The spread of the obtained values for each of the metrics can be observed in Figure 3. Each grey point depicts one observation, i.e. a value for a quality metric concerning a specific log and a specific model. The darker points in the figure indicate the mean value for each metric.

For the fitness metrics, it can be seen that the distributions of the observation are similarly distributed, save some minor exceptions. For example, there are no instances for which Token Based Fitness was lower than 0.125. Furthermore it is clear that the mass of the distribution for Behavioral Recall and Token Based Fitness is mostly close to one, while values for Alignment Based Fitness are slightly more uniformly spread.

Concerning the precision metrics, the mean values are rather different from one another, Behavioral Precision being a lot more pessimistic than Alignment Based Precision. Furthermore it can be seen that certain metrics have denser areas, with lots of observations, notably Alignment Based Precision and One Align Precision in the vicinity of 1. On the contrary, such dense areas do not exists for Behavioral Precision and Best Align precision.

Finally, the spread of values for the generalization metrics are quite different. Alignment Based Generalization has a left skewed distribution with most

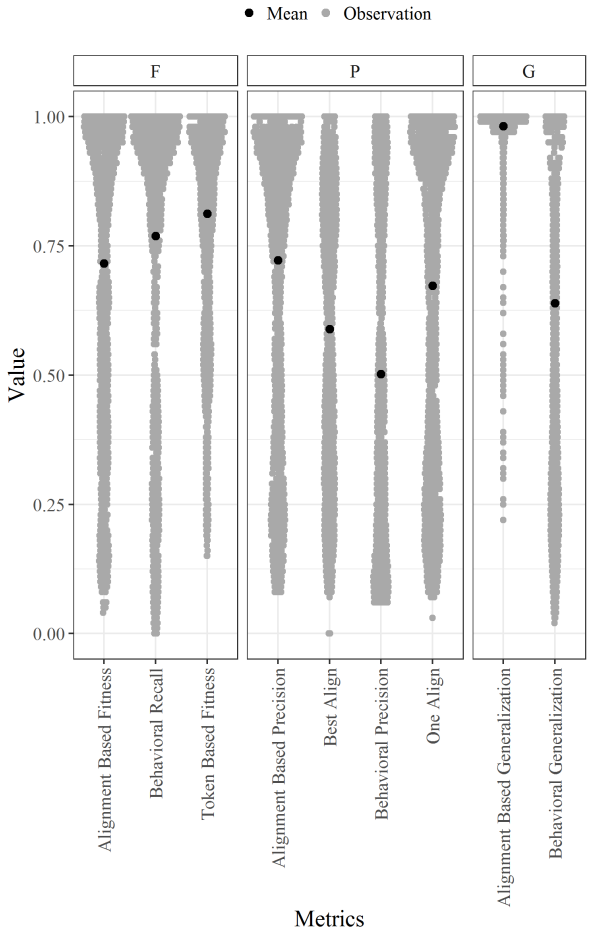


Figure 3: Distribution of values for different quality metrics

values close to 1. There are only a few values lower than 0.25. The spread of observations for Behavioral Generalization on the other hand does not contains gaps, and is much more evenly spread.

These first high level results indicate there are differences within each of the dimensions. However, to get a detailed view of their differences, one needs to connect all observations related to a specific log and model. In the next sections, additional insights are gained using correlation analysis and factor analysis.

4.2.1. Correlation analysis

In order to analyze the relations between the different metrics within and among dimensions, the correlation coefficients were computed, which are visualized in Figure 4. Some very interesting remarks can be made. When considering

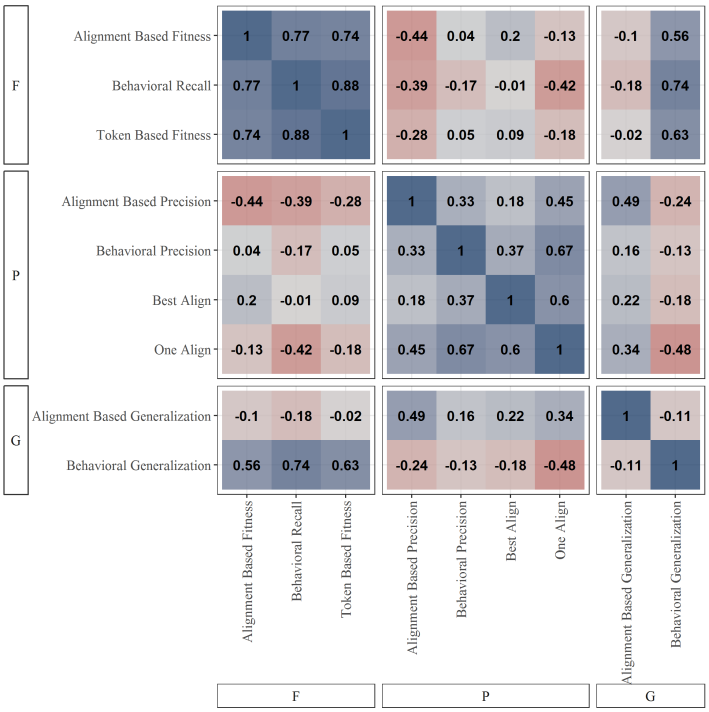


Figure 4: Correlation matrix.

metrics within each dimension, there is a clear difference between fitness and precision on the one hand, and generalization on the other hand. Firstly, it is very clear that all fitness metrics are highly correlated with each other, with for each pair a correlation higher than 0.80. The same is true for precision, although some of these correlations here are slightly lower. For generalization metrics, the situation is different however. Here, no relationship is found between the two metrics. The main reason for this is possibly the lack of variability for Alignment Based Generalization, as was already indicated in Figure 3. As a result, all correlation coefficients for this metric are close to zero.

When looking at relations across dimensions, two important results should be noted. Firstly, there are substantial negative correlations between fitness metrics and precision metrics. As such, models with a good fitness typically have a low precision, and vice versa. Secondly, Alignment Based Generalization is not correlated with either fitness or precision metrics, while Behavioral Generalization behaves like a fitness metric. Indeed, the latter is positively correlated with fitness metrics and negatively correlated with precision metrics.

Note that the negative correlation between fitness and precision metric is not necessarily a characteristic of the metrics itself. The conceptual analysis in Buijs [3] shows that both dimensions are theoretically independent from each

other. The negative correlations which were found are possibly related to the different process discovery algorithms.

In particular, a correlation analysis for each of the algorithms individually showed that for mostly positive correlations were found between fitness and precision metrics, except for the models discovered by the flower miner and to a lesser extent the alpha miner.

Figure 5 visualizes how discovered models are distributed in terms of their mean fitness and precision value, i.e. average over the different metrics. The saturation of the colors indicate where the mass of the discovered models is located for each algorithm. The colored lines represent a linear regression between mean fitness and mean precision for each of the algorithms.[2] The dashed line represents the negative linear regression for all miners combined, which is a result from the correlations in Figure 4.

The flower models were not the only reason to find an overall negative correlation, as this was still the case when the flowers models were omitted from Figure 4. Rather, it can be observed that it is the result of combining the search space of the different algorithms, which typically are slightly more focused towards either precision, or towards fitness. For instance, the alpha miner tends to find models which have a high precision, but a lower fitness, while the ILP miner finds models with the reverse characteristic. The combination of those leads to a perceived negative correlation. As a result, it can be stated that the fitness and precision dimensions are not negatively correlated per definition, which is in agreement with the theoretical foundations of the dimension. Rather, their relationship depends on which discovery algorithms are taken into consideration.

This analysis shows that the metrics that have been implemented tend to agree which each other within each dimensions, except for the generalization metrics. In the next paragraph, a factor analysis is conducted to delve further into these complex relationships.

4.2.2. Factor analysis

In order to further investigate the relationship between metrics within an across quality dimensions, an Exploratory Factor Analysis was done [14]. In order to decide on the number of factors to construct, a scree plot was composed to find the appropriate number of factors which explain the most variability in the data. this suggested that 2 or 3 factors would be most suitable. A factor analysis with 2 factors was chosen, based on the observation that a third factor did not have any significant loadings. As was stated in Section 3, a Promax rotation was used to increase the interpretability of the factors. As this is an oblique, non-orthogonal rotation, it allows for the fact that factors might be correlated.

[2]Note that is was not possible to draw a regression line for the Flower Miner, since each of these models had a fitness equal to one.

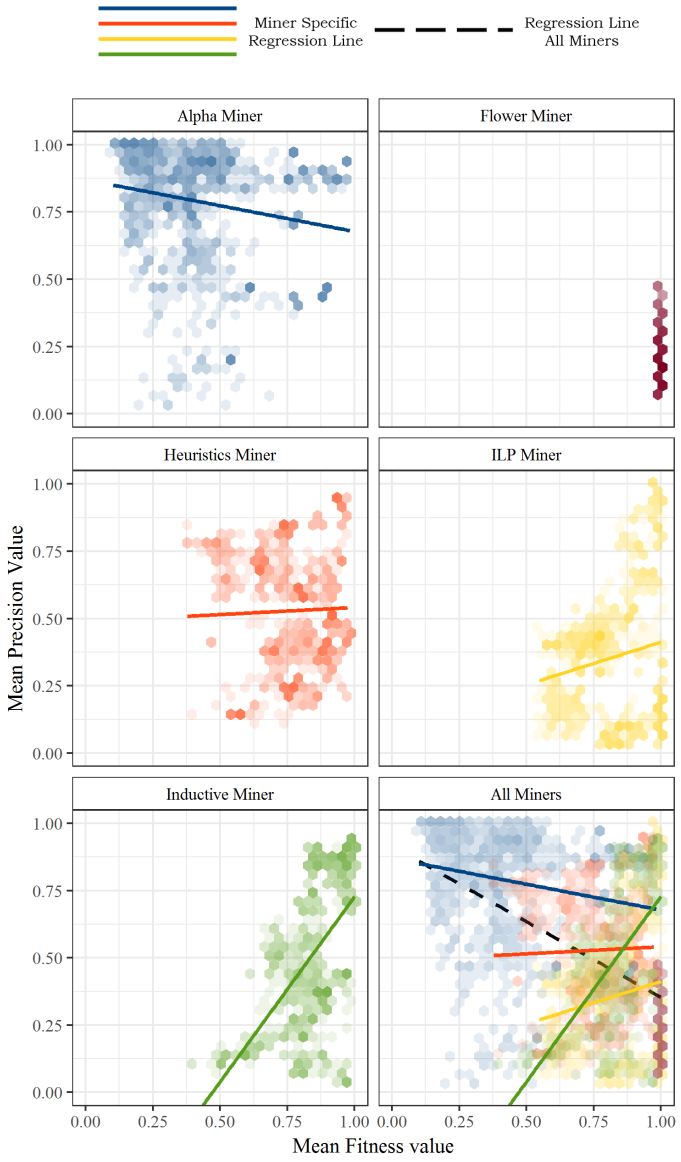


Figure 5: Relation between mean fitness and precision for different discovery algorithms, averaged over different fitness and precision metrics, respectively. The saturation of the color indicates the mass of the observations. Colored lines resemble the correlation between mean fitness and precision for each algorithm. The dashed lines resembles the correlations for all miners combined.

Table 5: Quality assessment factor analysis.

(a) Communality and MSA-value per metric

Metric	Communality	MSA
Alignment Based Fitness	0.7426	0.8221
Alignment Based Generalization	0.0085	0.0609
Alignment Based Precision	0.7260	0.7819
Best Align	0.7292	0.8481
Behavioral Generalization	0.7590	0.9112
Behavioral Precision	0.7154	0.8170
Behavioral Recall	0.9160	0.6994
One Align	0.9950	0.7906
Token Based Fitness	0.8920	0.7539

(b) Overall quality summary.

Characteristic	Value
Total Communality	0.7204
KMO	0.7838
RMSR	0.0370
Bartlett's p-value	0.0000

The quality of the factor analysis can be assessed using Table 5. The Kaiser-Meyer-Olkin statistic [14], which displays the proportion of variation between the different metrics, was equal to 0.7838, which is adequate. The Measures of Sampling Adequacy (MSA), which depict this proportion for each of the metrics individually are also quite high for most metrics. Only the Alignment Based Generalization has a remarkably low value for this metric. However, this does not pose problems, as the overall KMO value is high enough. The Root Mean Squared Residual is equal to 0.0370, and thereby well below the suggested maximum of 0.06 [21]. The Bartlett's test of Sphericity was done to test whether the correlation matrix was equal to a unity matrix, and thus factor analysis would be useless. However, this hypothesis was rejected with a p-value smaller than 0.0001.

The communalities for each of the specific metrics, shown in Table 5, show the proportion of variance for each of the metrics that is explained by the factor [14]. This shows that for the majority of the metrics more than 70% of the variance is explained by the factors. Also here, the Alignment Based Generalization is the only metric for which almost none of the variance is explained by the factors. Nonetheless, it can be concluded that the quality of the factor analysis is good and it is meaningful to interpret the factors.

The loadings of the factors that were found are shown in Figure 6 for each of the classical dimensions separately. It is clear that Factor 1 and Factor 2 represent the fitness and precision dimensions, respectively. All three fitness

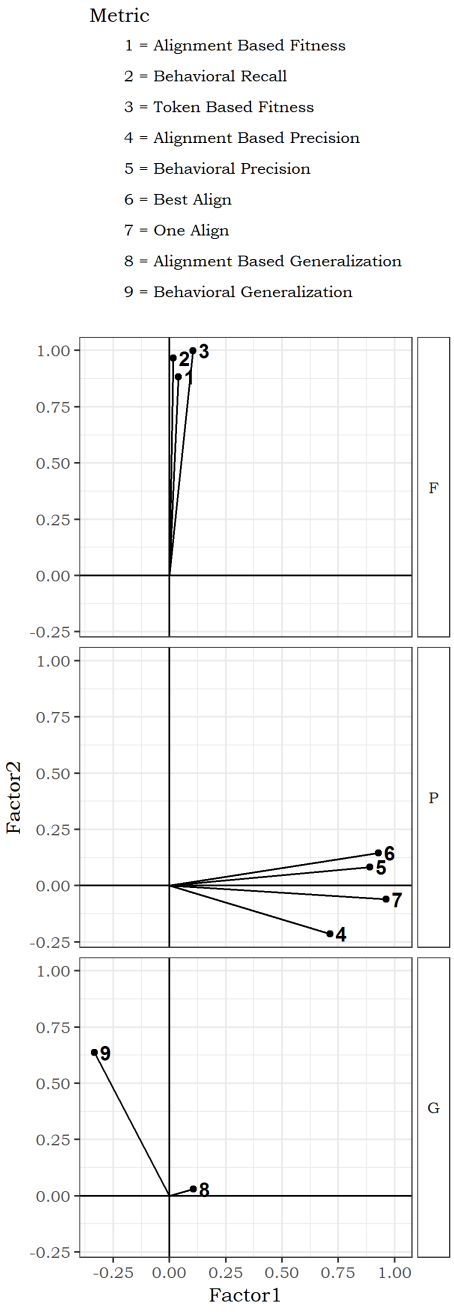


Figure 6: Factor loadings for a factor analysis with 3 factors and promax rotation.

metrics have a loading of more than 0.80 on the first factor. However, also the Behavioral Generalization metric has a considerably high loading on this factor. This means that, to a certain extent, it behaves in the same way as fitness metrics.

Subsequently, it can be seen that all precision metrics load reasonably high on Factor 2. As such, this factor seems to resemble the concept of precision. Behavioral Generalization is negatively loaded on this factor, but the loading is too small to attach any meaning onto it.

Furthermore, it is important to observe that Alignment Based Generalization did not have significant loadings on any of the factors, and this did not change when the number of factors was increased. This is striking, due to the fact that there is very little variance among the values obtained by this metric, as was shown in Figure 3. More specifically, the interquartile range is only 0.0028, between 0.9972 and 1.0000. As such, one would expect it to be very easy to explain a substantial part of the variance.

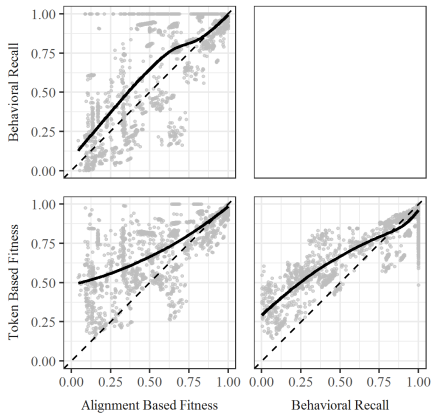
The fact that Behavioral Generalization has a high loading on the fitness-factor confirms the conclusion that was found earlier based on the correlation matrix. However, the relationship between fitness and generalization should not appear eccentric. A model with a good generalization is able to replay unobserved behavior. As a result, it appears logical that such a model can also replay observed behavior. The other way around, a model that cannot replay observed behavior, is unlikely to be able to replay unobserved but realistic behavior. The conceptual relationship between fitness and generalization is also discussed in Janssenswillen et al. [16], which stated that they are the same when the event log is noise-free and complete.

It can thus be concluded that both fitness metrics and precision metrics agree with each other, respectively. As a result, the validity of these metrics is approved. On the other hand, generalization metrics do not measure the same thing. The fact that one of the generalization metrics, i.e. Behavioral Generalization, loads reasonably high on the fitness-factor is expected to a certain extend. The Alignment Based Generalization metric seems to be a very insensitive metric, as the variance is very low.

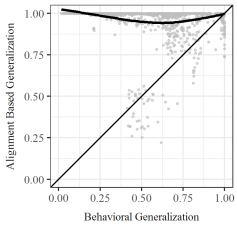
4.3. Sensitivity

The analysis of correlations and factors implicitly assume that the relations between different metrics are similar along the whole range, i.e. as well for models with a high quality as for models with a low quality. However, it is not impossible that some metrics tend to be more optimistic or more pessimistic. Moreover, metrics might undoubtedly agree on models with a very good or very bad fitness, but might judge models with intermediate fitness differently.

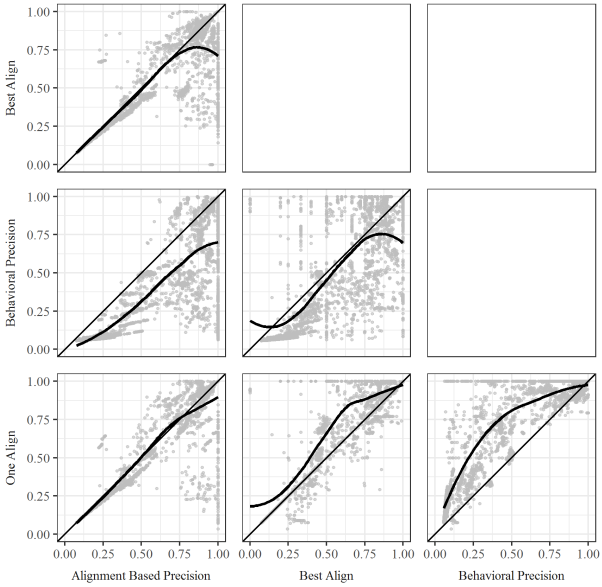
In order to examine the relationships between metrics on a more local level, scatter plots were drawn for each pair of metric in each dimension. Upon these, Lowess Smoothing lines were fitted [4]. The distance and difference in slope of the Lowess Smoothing in relationship with the diagonal line shows which of the two metric is more sensitive and more optimistic or pessimistic.



(a) Lowess Smoothings for Fitness.



(b) Lowess Smoothings for Generalization.



(c) Lowess Smoothings for Precision.

Figure 7: Lowess Smoothings for pairs of metrics within the dimensions Fitness, Precision and Generalization

In Figure 7a, Lowess Smoothing lines are shown which describe the relationships between the fitness metrics. When the smoothing line approximates the diagonal, the two metrics at hand score models equally. However, when the smoothing line falls below the diagonal, the y-as metric is more pessimistic.

When it sits above the diagonal, the y-as metric is more optimistic. Moreover, when the *direction* of the Lowess curve significantly differs from the diagonal, i.e. it is remarkably steep or flat, it can be said that there is a difference in sensitivity. I.e. when the Lowess curves turns toward a specific metric, as is the case in the lower left of Figure 7a, it can be said that this metric becomes less sensitive compared to the other metric.

In Figure 7a it can be seen that the smoothing line between Behavioral recall and Alignment Based Fitness is close to the diagonal, which indicates a good correspondence. When models have a higher fitness, Behavioral Recall and Alignment Based Fitness score models equally, while Behavioral Recall stays more optimistic as fitness decreases. Although Token Based Fitness and Alignment Based Fitness agree on models with perfect fitness, Token Based Fitness appears to be more optimistic than Alignment Based Fitness when the fitness of a model lowers. Token Based Fitness seems to be far less sensitive, as the gap between the Lowess curve and the diagonal increases when Alignment Based Fitness goes to zero. Moreover, Token Based Fitness also seems to be more optimistic than Behavioral Recall. However, for models with a good fitness, the two metrics correspond nearly perfect.

The same Lowess smoothing lines for precision metrics are shown in Figure 7c. Compared to Alignment Based Precision, Best Align and One Align have a perfect correspondence most of the time, although the latter are more pessimistic when models are scored very high by Alignment Based Precision. When this is the case, Best Align seems to be very insensitive, as the Lowess curve gets nearly horizontal. Behavioural Precision appears to score models more pessimistic on their preciseness compared to Alignment Based Precision in all of the cases.

Best Align correlates very well with One Align for almost all models, although One Align is slightly more optimistic. Compared to Behavioral Precision, Best Align score models equivalently when precision is moderate. However, when Best Align returns a high precision value, Behavioral Precision is less pessimistic and less sensitive. On the other hand, when Best Align scores the precision of a model to be very low, Behavioral Precision tends to be more optimistic. Finally, it can be observed that One Align returns more optimistic precision values than Behavioral Precision, except towards the extremes of the range.

At last, Figure 7b shows the relation between the two generalization metrics. In accordance with earlier results, most values for Alignment Based Generalization are in the vicinity of one. As a result, this metric is very insensitive and always more optimistic than Behavioral Generalization. However, the factor analysis shows that these metrics do not measure the same aspect in any case.

5. Conclusions and future work

In the context of process discovery, being able to evaluate the quality of obtained process models as a representation of the process at hand is essential. In order to do this, different quality dimensions were introduced and for each of the dimensions several metrics were implemented. However, only limited

empirical evidence exists on the behavior of these metrics and their relationships both within and across different quality dimensions. Nonetheless, the feasibility, validity and sensitivity of quality metrics are important aspects that need to be considered. In this paper, a large experiment was conducted in order to evaluate these characteristics.

It was found that, in general, metrics within the fitness and precision dimension, respectively, largely agree with each other. Nonetheless, several differences in sensitivity were found on a local scale which should certainly be taken into account when deciding on which metric to use. For instance, Token Based Fitness is for more insensitive when fitness is low and tends to be more optimistic compared to Behavioral Recall and Alignment Based Fitness. As to their feasibility, it can be said that all metrics which are based on alignments suffer from computational difficulties when the amount of behavior which can be replayed by the model is high. Based on this, it can be advised to use Behavioral Recall for measuring fitness. For measuring precision, One Align Precision is most suitable. Although it uses alignments and is therefore less slightly feasible, it is certainly more sensitive for differences in quality than its Behavioral counterpart.

No agreement was found between the generalization metrics taken into account. The Alignment Based Generalization was identified as a very insensitive metric with extremely little variance. On the other hand, the Behavioral Generalization appeared to have a strong relation with fitness, which can be corroborated with the conceptual definitions of the quality dimensions.

As most research is focused on the performance and effectiveness of process discovery algorithms, existing literature on the performance of quality metrics itself is limited. Although this paper only scratches the surface, it indicates that there is room for improvement and increased understanding in this area. Future research could be focused on the precise relationship between event log size or other event log characteristics and the feasibility of quality metrics. Moreover, the relation between generalization and fitness should be further investigated. In particular, it should be made clear when their metrics agree and when they do not, and whether this is according to their definition. Finally, further research is needed to find why certain metrics are more sensitive than others, and whether this related to certain characteristics of the behavior, for instance in terms of work-flow patterns.

Acknowledgments

The computational resources and services used in this work for both process discovery and process conformance tasks were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

[1] Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B. F., van der Aalst, W. M. P., 2012. Alignment based precision checking. In: Business Process Management Workshops. Springer, pp. 137–149.

[2] Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B. F., van der Aalst, W. M. P., 2015. Measuring precision of modeled behavior. Information Systems and e-Business Management, 1–31.

[3] Buijs, J. C. A. M., 2014. Flexible Evolutionary Algorithms for Mining Structured Process Models. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven.

[4] Cleveland, W. S., 1981. LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. The American Statistician 35 (1), 54–54.

[5] Cureton, E. E., Mulaik, S. A., 1975. The weighted varimax rotation and the promax rotation. Psychometrika 40 (2), 183–195.

[6] de Leoni, M., Maggi, F. M., van der Aalst, W. M. P., Jan. 2015. An alignment-based framework to check the conformance of declarative process models and to preprocess event-log data. Information Systems 47, 258–277, wOS:000344201100014.

[7] de Medeiros, A. K. A., 2006. Genetic process mining. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven.

[8] De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B., 2011. A critical evaluation study of model-log metrics in process discovery. In: Business Process Management Workshops. Springer, pp. 158–169, journal Club Marijke - 22/4/2015.

[9] De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B., 2012. A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. Information Systems 37 (7), 654–676.

[10] Fahland, D., van der Aalst, W. M. P., 2013. Simplifying discovered process models in a controlled manner. Information Systems 38 (4), 585–605.

[11] Garcia-Banuelos, L., Dumas, M., La Rosa, M., De Weerdt, J., Ekanayake, C. C., Dec. 2014. Controlled automated discovery of collections of business process models. Information Systems 46, 85–101, wOS:000340318800006.

[12] Goedertier, S., Martens, D., Vanthienen, J., Baesens, B., 2009. Robust process discovery with artificial negative events. The Journal of Machine Learning Research 10, 1305–1340.

[13] Greco, G., Guzzo, A., Ponieri, L., Sacca, D., 2006. Discovering expressive process models by clustering log traces. Knowledge and Data Engineering, IEEE Transactions on 18 (8), 1010–1027.

[14] Hair, J. F., Black, Babin, Anderson, 2013. Multivariate Data Analysis. Pearson Education Limited, Harlow, oCLC: 857714899.

[15] Janssenswillen, G., Depaire, B., Jouck, T., 2016. Calculating the number of unique paths in a block-structured process model. In: Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data 2016.

[16] Janssenswillen, G., Jouck, T., Creemers, M., Depaire, B., Sep. 2016. Measuring the Quality of Models with Respect to the Underlying System: An Empirical Study. In: Rosa, M. L., Loos, P., Pastor, O. (Eds.), Business Process Management. No. 9850 in Lecture Notes in Computer Science. Springer International Publishing, pp. 73–89.

[17] Jouck, T., Depaire, B., Mar. 2016. Generating Artificial Data for Empirical Analysis of Process Discovery Algorithms: A Process Tree and Log Generator. Technical report, Universiteit Hasselt, Universiteit Hasselt.

[18] Leemans, S. J. J., Fahland, D., van der Aalst, W. M. P., 2013. Discovering block-structured process models from event logs-a constructive approach. In: Application and Theory of Petri Nets and Concurrency. Springer, pp. 311–329.

[19] Mendling, J., Neumann, G., Van Der Aalst, W., 2007. Understanding the occurrence of errors in process models based on metrics. In: On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS. Springer, pp. 113–130.

[20] Muñoz-Gama, J., Carmona, J., 2010. A fresh look at precision in process conformance. In: Business Process Management. Vol. 6336. Springer, Hoboken, NJ, USA, pp. 211–226.

[21] Preacher, K. J., MacCallum, R. C., 2002. Exploratory factor analysis in behavior genetics research: Factor recovery with small sample sizes. Behavior genetics 32 (2), 153–161.

[22] Rozinat, A., De Medeiros, A. K. A., Günther, C. W., Weijters, A. J. M. M., Van der Aalst, W. M. P., 2007. Towards an Evaluation Framework for Process Mining Algorithms. Beta, Research School for Operations Management and Logistics.

[23] Rozinat, A., van der Aalst, W. M. P., 2008. Conformance checking of processes based on monitoring real behavior. Information Systems 33 (1), 64–95.

[24] Schrepfer, M., Wolf, J., Mendling, J., Reijers, H. A., 2009. The impact of secondary notation on process model understanding. In: IFIP Working Conference on The Practice of Enterprise Modeling. Springer, pp. 161–175.

[25] Senderovich, A., Weidlich, M., Yedidsion, L., Gal, A., Mandelbaum, A., Kadish, S., Bunnell, C. A., Dec. 2016. Conformance checking and performance improvement in scheduled processes: A queueing-network perspective. Information Systems 62, 185–206, wOS:000384780500011.

[26] Van der Aalst, W. M. P., Adriansyah, A., van Dongen, B., 2012. Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2 (2), 182–192.

[27] van der Aalst, W. M. P., Reijers, H. A., Weijters, A. J., van Dongen, B. F., De Medeiros, A. A., Song, M., Verbeek, H. M. W., 2007. Business process mining: An industrial application. Information Systems 32 (5), 713–732.

[28] van der Aalst, W. M. P., Weijters, T., Maruster, L., 2004. Workflow mining: Discovering process models from event logs. Knowledge and Data Engineering, IEEE Transactions on 16 (9), 1128–1142.

[29] Van der Werf, J. M. E., van Dongen, B. F., Hurkens, C. A., Serebrenik, A., 2008. Process discovery using integer linear programming. In: Applications and Theory of Petri Nets. Springer, pp. 368–387.

[30] vanden Broucke, S. K., Delvaux, C., Freitas, J., Rogova, T., Vanthienen, J., Baesens, B., 2014. Uncovering the relationship between event log characteristics and process discovery techniques. In: Business Process Management Workshops. Springer, pp. 41–53.

[31] vanden Broucke, S. K. L. M., De Weerdt, J., Vanthienen, Jan, B., Baesens, B., 2014. Determining process model precision and generalization with weighted artificial negative events. Knowledge and Data Engineering, IEEE Transactions on 26 (8), 1877–1889.

[32] vanden Broucke, S. K. L. M., De Weerdt, J., Vanthienen, J., Baesens, B., 2013. A Comprehensive Benchmarking Framework (CoBeFra) for conformance analysis between procedural process models and event logs in ProM. In: Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium On. IEEE, pp. 254–261.

[33] Weber, P., Bordbar, B., Tino, P., Majeed, B., 2011. A framework for comparing process mining algorithms. In: GCC Conference and Exhibition. IEEE, pp. 625–628.

[34] Weidlich, M., Polyvyanyy, A., Desai, N., Mendling, J., Weske, M., 2011. Process compliance analysis based on behavioural profiles. Information Systems 36 (7), 1009–1025.

[35] Weijters, A. J. M. M., van Der Aalst, W. M. P., De Medeiros, A. K. A., 2006. Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP 166, 1–34.