Improving structural medical process comparison by exploiting domain knowledge and mined information

Stefania Montani\textsuperscript{a,}\textsuperscript{*}, Giorgio Leonardi\textsuperscript{a}, Silvana Quaglini\textsuperscript{b}, Anna Cavallini\textsuperscript{c}, Giuseppe Micieli\textsuperscript{c}

\textsuperscript{a} Dipartimento di Scienze e Innovazione Tecnologica, Computer Science Institute, Università del Piemonte Orientale, Viale Michel 11, I-15121 Alessandria, Italy
\textsuperscript{b} Department of Electrical, Computer and Biomedical Engineering, Università di Pavia, Via Ferrata 1, I-27100 Pavia, Italy
\textsuperscript{c} Istituto di Ricerca e Cura a Carattere Scientifico Fondazione “C. Mondino”, Via Mondino 2, I-27100 Pavia, Italy

\textbf{ARTICLE INFO}

\textbf{Article history:}
Received 26 June 2013
Received in revised form 12 May 2014
Accepted 12 July 2014

\textbf{Keywords:}
Process mining and comparison
Graph edit distance
Stroke management

\textbf{ABSTRACT}

\textbf{Objectives:} Process model comparison and similar process retrieval is a key issue to be addressed in many real-world situations, and a particularly relevant one in medical applications, where similarity quantification can be exploited to accomplish goals such as conformance checking, local process adaptation analysis, and hospital ranking. In this paper, we present a framework that allows the user to: (i) mine the actual process model from a database of process execution traces available at a given hospital; and (ii) compare (mined) process models. The tool is currently being applied in stroke management.

\textbf{Methods:} Our framework relies on process mining to extract process-related information (i.e., process models) from data. As for process comparison, we have modified a state-of-the-art structural similarity metric by exploiting: (i) domain knowledge; (ii) process mining outputs and statistical temporal information. These changes were meant to make the metric more suited to the medical domain.

\textbf{Results:} Experimental results showed that our metric outperforms the original one, and generated output closer than that provided by a stroke management expert. In particular, our metric correctly rated 11 out of 15 mined hospital models with respect to a given query. On the other hand, the original metric correctly rated only 7 out of 15 models. The experiments also showed that the framework can support stroke management experts in answering key research questions: in particular, average patient improvement decreased as the distance (according to our metric) from the top level hospital process model increased.

\textbf{Conclusions:} The paper shows that process mining and process comparison, through a similarity metric tailored to medical applications, can be applied successfully to clinical data to gain a better understanding of different medical processes adopted by different hospitals, and of their impact on clinical outcomes. In the future, we plan to make our metric even more general and efficient, by explicitly considering various methodological and technological extensions. We will also test the framework in different domains.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Process model comparison and similar process retrieval is a key issue to be addressed in many real-world situations. For example, when two companies are merged, process engineers need to compare processes originating from the two companies, in order to analyze their possible overlaps, and to identify areas for consolidation. Moreover, large companies build over time huge process model repositories, which serve as a knowledge base for their ongoing process management/enhancement efforts. Before adding a new process model to the repository, process engineers have to check that a similar model does not already exist, in order to prevent duplication. Particularly interesting is the case of medical process model comparison, where similarity quantification can also be exploited in a conformance checking perspective. Indeed, the process model actually implemented at a given healthcare organization can be compared to the existing reference clinical guideline, to check conformance, and/or to understand the level of adaptation to local constraints that may have been required. As a matter of fact, the existence of local resource constraints may lead to
differences between the models implemented at different hospitals, even when referring to the treatment of the same disease (and to the same guideline). A quantification of these differences (and maybe a ranking of the hospitals derived from it) can be exploited for several purposes, like, e.g., administrative purposes, performance evaluation and public funding distribution. The actual medical process models are not always explicitly available at the healthcare organization. However, a database of process execution traces (also called the “event log”) can often be reconstructed starting from data that hospitals collect through their information systems (in the best case by means of workflow technology).

In this case, process mining techniques [1] can be exploited, to extract process models from event log data. Stemming from these considerations, in this work we present a framework, which allows the user to:

1. extract the actual process model from the available medical process execution traces, through process mining techniques;
2. perform medical process model comparison, to fulfill the objectives described above.

Item 2 has required the introduction of proper metrics, in order to quantify process model similarity. We could rely on an extensive literature when studying this topic (see Section 4). In particular, since process mining extracts the process model in the form of a graph, our work is located in the research stream on graph structural similarity, and on graph edit distance-based approaches [2,3]. The state of the art on structural similarity on process models is represented by the work by Dijkman et al. [2]. Specifically, we have extended the work in [2], by:

- exploiting domain knowledge;
- exploiting process mining outputs and statistical temporal information.

We believe that the use of domain knowledge represents a significant enhancement in the metric definition, which, otherwise, would operate in a “blind” and context-independent fashion. Indeed, the original metric in [2] is completely independent of the domain of application. On the other hand, when domain knowledge is available, rich and well consolidated, as is often the case in medicine, its exploitation can surely improve the quality of any automated support to the expert’s work – including process comparison (see e.g., [4]). Moreover, the use of additional information extracted from data, and of temporal information in particular, can be a relevant advancement as well, in fields in which the role of time can be very critical, like, e.g., emergency medicine. We are currently applying our framework to stroke management. In this domain, the positive experimental results we have obtained support the statements above. Indeed, our metric has proved to outperform the original metric in [2], and to generate outputs that are closer to those provided by a stroke management expert (see Section 3.1). Having verified the reliability of our tool through the experimental study described in Section 3.1, we have then applied it to address a key, open research question in stroke management, namely: do similar process models (implemented in different stroke units) lead to similar clinical outcomes (e.g., patient survival rate and/or patient improvement rate at discharge)? Some interesting conclusions on this issue were obtained (see Section 3.2), testifying the potential clinical usefulness of our contribution. The paper is organized as follows. Section 2 provides the details of our methodological approach. Section 3 showcases experimental results. Section 4 compares our contribution to related works. Section 5 shows the limitations of our work, and our future research directions, meant to overcome the open issues. Section 6 illustrates our concluding remarks.

2. Methods

In this section, we will first introduce process mining and the ProM tool; then we will provide the technical details of our metric.

2.1. Process mining and the ProM tool

Process mining describes a family of a-posteriori analysis techniques exploiting the information recorded in event logs, to extract process-related information (e.g., process models). Typically, these approaches assume that it is possible to sequentially record events such that each event refers to an activity (i.e., a well-defined step in the process) and is related to a particular process instance. Furthermore, some mining techniques use additional information such as the timestamp of the event, or data elements recorded with the event.

Traditionally, process mining has focused on discovery, i.e., deriving process models and execution properties from event logs. It is important to mention that there is no a-priori model, but, based on logs, some models, e.g., a Petri net, are constructed. However, process mining is not limited to process models (i.e., control flow), and recent process mining techniques have focused more and more on other perspectives, e.g., the organizational perspective, the performance perspective or the data perspective. Moreover, as clearly stated in [5], process mining also supports conformance analysis and process enhancement. In this paper, however, we will focus on the process perspective.

In our approach, we resorted to the process mining tool called ProM, extensively described in [6], and to ProM’s Heuristic Miner [7] plug-in. ProM (and specifically its newest version ProM 6) is a platform-independent open source framework that supports a wide variety of process mining and data mining techniques, and can be extended by adding new functionalities in the form of plug-ins. Heuristic Miner [7] is a plug-in for mining process models from event logs. Heuristic Miner receives in input the log, and considers the order of the events within every single process instance execution. The timestamp of an activity is used to calculate this ordering. Heuristic Miner can be used to express the main behavior (i.e., not all details) registered in a log. Indeed, abstract information, such as the presence of composite tasks (i.e., tasks semantically related to their constituent activities by means of the “part-of” relation), cannot be derived by Heuristic Miner, which will only build a model including ground (i.e., not further decomposable) activities. On the other hand, it can mine the presence of short-distance and long-distance dependencies (i.e., direct or indirect sequence of activities), and information about parallelism, with a certain degree of reliability (see also Section 2.2). The output of the mining process is provided as a graph, known as the “dependency graph”, where nodes represent activities, and edges represent control flow information. Heuristic miner does not extract behavioral/cause relationships. The output can be converted into other formalisms as well. Currently, we have chosen to rely on Heuristics Miner, because it is known to be tolerant to noise, a problem that may affect medical event logs (e.g., sometimes the logging may be incomplete). Testing of other mining algorithms available in ProM 6 is, however, foreseen in our future work, as discussed in Section 5.

2.2. Distance definition for process model comparison

In order to compare process models on the basis of their distance, we have introduced a distance definition that extends previous literature contributions [2,3] (see also Section 4) by properly considering the information mined/learned from data, as well as
the available domain semantic knowledge. Remarkably, the contribution in [2] represents the state of the art, when dealing with structural similarity between process models. In particular, since mined process models are represented in the form of graphs (where nodes represent activities and edges provide information about the control flow), we define a distance based on the notion of graph edit distance [3]. Such a notion calculates the minimal cost of transforming one graph into another by applying edit operations i.e., insertions/deletions and substitutions of nodes, and insertions/deletions of edges. While string edit distance looks for an alignment that minimizes the cost of transforming one string into another by means of edit operations, in graph edit distance we have to look for a mapping. A mapping is a function that matches nodes to nodes, and edges to edges. Among all possible mappings, we will select the one that leads to the minimal cost, having properly quantified the cost of every type of edit operation. As in [2], we provide a normalized version of the approach in [3]. Moreover, with respect to [2], we introduce two novel contributions:

1. We calculate the cost of node substitution $f_{subn}$ (see Definition 4 in Section 2.2.2) by applying Palmer’s “taxonomic distance” [8] (see Definition 1 in Section 2.2.2), and not string edit distance on node names as in [2]. The use of this definition allows us to explicitly take into account semantic domain knowledge;

2. We add a cost contribution related to edge substitution ($f_{sube}$ in Definition 4), which incorporates information learned from the event log, namely: (i) the percentage of patients that have followed a given edge, (ii) the reliability of a given edge, i.e., of the control flow relationship between two activities, and (iii) statistics about the temporal duration of a given edge. Both items (i) and (ii) are outputs of Heuristic Miner (see Definitions 2 and 3 in Section 2.2.2). As for item (iii), we have directly calculated the mean and the standard deviation of the temporal duration of edges, by referring to the content of the event log.

The rationale underlying the first contribution, and thus the use of taxonomic distance, is that we believe that the use of semantic domain knowledge represents a significant enhancement in the metric definition, which, otherwise, would operate in a “blind” and context-independent fashion. Indeed, the original metric in [2] is completely independent of the domain of application. On the other hand, in medicine domain knowledge is typically available, and can be resorted to, in order to tailor the metric to the characteristics of the application. This can enhance usability and usefulness in practice. While the approach in [2] relies on string edit distance on node names to compare two activities (i.e., two nodes), in [9] more complex proposals were presented. One of these, called “type similarity,” allows a type to be assigned to every node, and then types to be compared. Quite simply, identical types have a distance of 0, while different types have a distance of 1. Type similarity has proved to outperform all other node comparison metrics in [9].

Starting from this observation, we refine the concept of type, and, by introducing a taxonomic structure, we introduce a hierarchy of types (more properly called “classes” below), whose leaves are the activities that can be found in the process models. Taxonomic distance [8] then allows us to exploit the hierarchical structure, since the distance between two activities is set to the normalized number of edges on the path between the two activities themselves in the taxonomy (see Definition 1). In our specific field, classes are defined on the basis of their goal. Thus, the underlying idea of the use of taxonomic distance in the case of medical processes is that two different activities are more or less distant on the basis of their goal. As for the choice of relying on Palmer’s distance [8], it can be observed that it is one of the most representative metrics based on the ontology graph structure, and it has proved to provide very good performance results in practice (see e.g., [10]). However, it is worth noting that our framework is modular, and a different metric could be used to calculate the cost of node substitution in $f_{subn}$ without altering the overall approach.

In fact, time can be a very important parameter in medical application, particularly when referring to emergency medicine, as it is in the case of stroke. As we will see in the formal definitions, a set of weights will allow us to tune the importance of the various contributions, making the approach easily adaptable to various medical applications and experimental needs. In Section 2.2.1, we will provide a description of the taxonomy we are using. In Section 2.2.2, we will formally provide the metrics definitions.

2.2.1. Stroke management taxonomy

Our co-author A.C., an experienced physician in stroke patient management, has provided us with the domain knowledge to define the taxonomy partially reported in Fig. 1. The taxonomy, which was developed by using the Protégé ontology editor2, is composed of 111 classes, organized in a hierarchy of six levels.

First, the taxonomy divides the activities into two main classes: activities that take place in the emergency phase (EM – generally performed in the emergency room), and activities that take place during the hospitalization phase (H – generally performed in the stroke unit) – see also Section 3. These two main classes correspond to the two main goals, which are: (1) to face the stroke emergency as quickly as possible, and (2) to plan the patient’s monitoring and secondary prevention. Moreover, these classes are further refined in subclasses, according to more specific goals. This refinement continues to the leaves, where the most specific activities are represented.

Some subclasses (e.g., diagnostic procedures and therapy) are repeated in both main classes, but their goal is very different: for example, a computerized tomography (CT) or a magnetic resonance (MR) in the emergency phase have the main goal of excluding a hemorrhagic stroke, while the same examinations in the hospitalization phase are performed to monitor stroke evolution and refine the etiopathogenetic diagnosis. Thus, the activities “H brain CT” and “H brain MR” (see Fig. 1), which are put together in the brain parenchyma evaluation node of the hospitalization (H) phase, although based on very different technologies, are closer than “H brain CT” and “EM brain CT”, because these are executed to investigate brain parenchyma in the two different EM and H phases. The organization of the taxonomy also makes the distance between “H brain CT” and “H brain MR” smaller than the one between “H brain CT” and “H transthoracic echocardiogram” because, even if these last two activities are performed in the same phase (hospitalization), their goal is completely different (monitoring changes in the brain parenchyma vs. cardiologic diagnosis).

Definition 1 in the next section quantifies these differences.

2.2.2. Metrics

In this section, we formally provide our definitions for distance calculation, together with a simple application example.

---

2 http://protege.stanford.edu [accessed on 04.11.13]
**Definition 1. Taxonomic distance.** Let \( \alpha \) and \( \beta \) be two activities in the taxonomy \( T \), and let \( \gamma \) be the closest common ancestor of \( \alpha \) and \( \beta \). The taxonomic distance \( dT(\alpha, \beta) \) between \( \alpha \) and \( \beta \) is defined as [8]:

\[
dT(\alpha, \beta) = \frac{N_1 + N_2}{N_1 + N_2 + 2 \times N_3}
\]

where \( N_1 \) is the number of edges in the path from \( \alpha \) and \( \gamma \) in \( T \), \( N_2 \) is the number of edges in the path from \( \beta \) and \( \gamma \), and \( N_3 \) is the number of edges in the path from the taxonomy root and \( \gamma \).

**Definition 2. Reliability.** The reliability of the edge \( e_i \) assessing that activity \( a \) follows activity \( b \) in a direct sequence (i.e., \( e_i \) is an arc from \( b \) to \( a \)) is calculated as [7]:

\[
\text{rel}(e_i) = \frac{|a > b| - |b > a|}{|a > b| + |b > a| + 1}
\]

where \(|a > b|\) is the number of occurrences in which activity \( a \) directly follows activity \( b \) in the event log, and \(|b > a|\) is the number of occurrences in which activity \( b \) directly follows activity \( a \).

A negative reliability value means that we must conclude that the opposite pattern holds, i.e., activity \( b \) follows activity \( a \). Indeed, the reliability of a relationship (e.g., activity \( a \) follows activity \( b \)) is not only influenced by the number of occurrences of this pattern in the log, but it is also (negatively) determined by the number of occurrences of the opposite pattern (\( b \) follows \( a \)). However, edges with a negative reliability will not appear in the dependency graph (due to threshold mechanisms and proper heuristics [7], which rule them out). Therefore, we will deal with reliability values \( \epsilon(0, 1) \).

**Definition 3. Percentage of patients.** The percentage of patients that crossed edge \( e_i \), assessing that activity \( a \) follows activity \( b \) in a direct sequence, is calculated as:

\[
\text{pat}(e_i) = \frac{|a > b|}{\text{ALLTRACE}}
\]

where \(|a > b|\) is the number of traces in which activity \( a \) directly follows activity \( b \) in the event log, and \( \text{ALLTRACE} \) is the total number of available traces (i.e., of patients) in the event log.

With this definition, percentage is scaled \( \in [0, 1] \).

**Definition 4. Extended graph edit distance.** Let \( G_1 = (N_1, E_1) \) and \( G_2 = (N_2, E_2) \) be two graphs, where \( E_1 \) and \( N_1 \) represent the sets of edges and nodes of graph \( G_1 \). Let \( M \) be a partial injective mapping \([1] \) that maps nodes in \( N_1 \) to nodes in \( N_2 \) and let \( 
- \) be the sets of substituted nodes, inserted or deleted nodes and inserted or deleted edges with respect to \( M \). In particular, a substituted edge connects a pair of substituted nodes in \( M \). The fraction of inserted or deleted nodes, denoted by \( f_{skipn} \), the fraction of inserted or deleted edges, denoted by \( f_{skipne} \), and the average distance of substituted nodes, denoted by \( f_{subn} \), are defined as follows:

\[
\text{f_{skipn}} = \frac{|\text{skipn}|}{|N_1| + |N_2|}
\]

\[
\text{f_{skipne}} = \frac{|\text{skipne}|}{|E_1| + |E_2|}
\]

\[
\text{f_{subn}} = \frac{2 \times \sum_{n, m \in M} dT(n, m)}{|\text{subn}|}
\]

where \( n \) and \( m \) are two mapped nodes in \( M \), and \( dT(n, m) \) is defined as in **Definition 1**. The average distance of substituted edges \( f_{sube} \) is defined as follows:

\[
\text{f_{sube}} = \frac{2 \times \sum_{(n_1, m_2), (m_1, m_2) \in M} (\text{rel}(e_1) - \text{rel}(e_2)) + |\text{pat}(e_1) - \text{pat}(e_2)| + |\text{mt}(e_1) - \text{mt}(e_2)| + |\text{st}(e_1) - \text{st}(e_2)|}{|\text{sube}|}
\]

where \( e_1 \) (connecting node \( n_1 \) to node \( m_1 \)) and \( e_2 \) (connecting node \( n_2 \) to node \( m_2 \)) are two substituted edges in \( M \); \( \text{rel}(e_i) \) is the reliability of edge \( e_i \) (see **Definition 2**); \( \text{pat}(e_i) \) is the percentage of patients that crossed edge \( e_i \) (see **Definition 3**); \( \text{mt}(e_i) \) and \( \text{st}(e_i) \) are statistical values (mean and standard deviation of the elapsed times) calculated over all the occurrences of the \( m_i > n_i \) pattern in the traces of hospital \( i \), and normalized in \([0, 1]\) dividing by the duration of the longest \( m_i > n_i \) pattern in the log.  

![Fig. 1. An excerpt from the domain taxonomy.](image_url)
The extended graph edit distance induced by the mapping $M$ is:

$$
\text{ext}_{ed}(M) = \frac{\sum w_{subn} + f_{skipe} + w_{skip} \times f_{skipe} + w_{sub} \times f_{sub} + w_{sub} \times f_{sube} + w_{subn} \times f_{sube}}{w_{skip} + w_{skip} + w_{sub} + w_{subn}}
$$

where $w_{subn}$, $w_{sub}$, $w_{skip}$ and $w_{skipe}$ are proper weights $e[0, 1]$.

Note that $rel(ei)$, $pat(ei)$, $mt(ei)$ and $st(ei)$ could be combined in a weighted average as well. In our experiments, according to our medical co-author A.C.'s opinion, weights were set as follows: $w_{subn} = 1$; $w_{sub} = 0.2$; $w_{skip} = 1$; $w_{skipe} = 0.6$. The rationale behind this choice is the following: in stroke management, activities (i.e., graph nodes) are more important than control flow, therefore a node substitution (see $w_{subn}$) or deletion (see $w_{skip}$) have the highest weights. Edge deletion (see $w_{skipe}$) is more important than edge substitution (see $w_{sub}$), because a change in the activity execution sequence must be strongly penalized (even if not as strongly as a change in the activities themselves). The penalty for edge substitution is the lowest, because it refers to situations in which the activity sequence is identical in the models; only, information (e.g., reliabilities or times) associated to the edges at hand may be different. It is important to take into account these differences, but they don’t impact as much as a change in the model control flow. It is worth noting that a sensitivity analysis can be conducted to automate weight setting, when expert knowledge is not available.

The extended graph edit distance of two graphs is the minimal possible distance induced by a mapping between these graphs.

It can be easily verified that our metric, being an extension of the edit distance, preserves the metric properties of non-negativity, identity of indiscernibles, and symmetry. Some versions of the normalized edit distance may fail the triangle inequality in a few very specific experimental situations (see [11]), but the problem can be tackled, as discussed in [12]. Moreover, as clearly stated in [13], triangle inequality is not considered to be essential for measuring process distance.

To find the mapping that leads to the minimal distance we resort to a greedy approach, as described in [2], in order to contain computational costs. It can be shown that the algorithm works in cubic time on the number of nodes of the larger graph [2]. As is well known, a greedy algorithm is an algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage, with the hope of finding a global optimum. A greedy strategy does not in general produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time. Fig. 2 shows two simple graphs (named $G_1$ and $G_2$), which will be used to illustrate how distance calculation takes place according to our definitions. Fig. 3 presents a taxonomy, which will be used to calculate taxonomic distance (see Definition 1) in the example graphs.

In the graphs in Fig. 2, nodes (boxes) represent the activities, while edges represent sequence control flow information. The labels on each edge provide, from top to bottom: (1) reliability of the edge ($rel(ei)$, see Definition 2), (2) percentage of patients crossing the edge ($pat(ei)$, see Definition 3), (3) mean ($mt(ei)$, see Definition 4) and (4) standard deviation ($st(ei)$, see Definition 4) of the elapsed time between the two sequential activities connected by the edge $ei$.

As already stated, the extended graph edit distance of two graphs is the minimal distance induced by a mapping between the graphs $G_1$ and $G_2$. The mapping identifies what parts of the two graphs can be overlapped through substitution operations. In Fig. 2, the activities and edges that are substituted from one graph to the other, according to the proposed mapping, are highlighted by a thicker line style. The mapping inducing the minimal extended graph edit distance between the graphs $G_1$ and $G_2$ is the following one:

$$
\text{MAP}(G_1, G_2) = \langle M, N \rangle, \langle J, J \rangle, \langle O, O \rangle, \langle K, K \rangle, \langle S, T \rangle, \langle Y, X \rangle, \langle U, V \rangle
$$

The set of substituted edges and the set of inserted/deleted nodes and edges can be inferred from the mapping itself. This mapping contains three pairs of identical nodes, appearing in both $G_1$ and $G_2$, i.e., $(J, J)$, $(O, O)$, and $(K, K)$, but also four pairs whose taxonomic distance is sufficiently low to be considered a convenient substitution. Globally, 14 nodes are substituted, i.e., $|subn| = 14$ (see Definition 4).

In the mapping, the activity $M$ in $G_1$ is substituted by the activity $N$ in $G_2$. Indeed: (1) $M$ does not appear in $G_2$, but (2) $N$ and $M$ have a low taxonomic distance. In fact (see Fig. 3), $M$ and $N$ are siblings and have $E$ as a common parent. Referring to Definition 1, their taxonomic distance is:

$$
td(M, N) = \frac{1 + 1}{1 + 1 + 2 \times 2} = 0.33
$$

For the same reason, the mapping contains the pairs $(Y, X)$, $(U, V)$ and $(S, T)$, whose taxonomic distance can be calculated analogously. Obviously, the taxonomic distance of the pairs $(J, J)$, $(O, O)$, and $(K, K)$ is 0.

The contribution of $f_{subn}$ is then obtained by applying the formula in Definition 4:

$$
\text{f}_{subn} = 2 \times (0.33 + 0.33 + 0 + 0 + 0.33 + 0.33) = 0.18
$$

To calculate $f_{skip}$ (see Definition 4), we need to calculate the number of inserted/deleted nodes in the two graphs. These nodes are: $P$ in $G_1$, and $Q, Z$ and $R$ in $G_2$. Thus, $|skip| = 1 + 3 + 4$, $|N_1| = 8$ and $|N_2| = 10$, leading to:

$$
\text{f}_{skip} = \frac{4}{8 + 10} = 0.22
$$

The calculation of $f_{skipe}$ (see Definition 4) is similar. The inserted/deleted edges are as follows: $(S, P)$ and $(P, Y)$ in $G_1$, and $(T, Q)$, $(T, Z)$, $(Q, R)$, $(Z, R)$ and $(R, X)$ in $G_2$. Therefore $|skipe| = 5 + 2 = 7$, $|E_1| = 8$ and $|E_2| = 11$, leading to:

$$
\text{f}_{skipe} = \frac{7}{8 + 11} = 0.36
$$

The calculation of $f_{sube}$ proceeds as follows (see Definition 4). For each pair of substituted edges $ei$ and $ej$, it is necessary to compare the labels $rel(ei)$ and $rel(ej)$, $pat(ei)$ and $pat(ej)$, $mt(ei)$ and $mt(ej)$, $st(ei)$ and $st(ej)$. In the example, the edge $(M, J)$ in $G_1$ is replaced with the edge $(N, J)$ in $G_2$. $(J, O)$ is replaced with $(J, O)$, $(J, S)$ with $(J, T)$, $(O, K)$ with $(O, K)$, $(K, Y)$ with $(K, X)$ and $(Y, U)$ with $(X, V)$. The number of substituted edges is therefore $|sube| = 12$. Considering the replacement $(M, J)$ with $(N, J)$, the comparison of their labels provides the following value:

$$
\text{comp}((M, J), (N, J)) = (|0.993 - 0.997| + |1.0 - 1.0| + |0.07 - 0.01| + |0.11 - 0.01|) = 0.164
$$

The label comparison for all the other substituted edges is calculated similarly. Finally, the contribution $f_{sube}$ assumes the value:

$$
f_{sube} = \frac{2 \times (0.164 + 0.193 + 0.377 + 0.382 + 0.958 + 1.764)}{4 \times 12} = 0.159
$$
Fig. 2. Two simple graphs.

The overall distance between the graphs $G_1$ and $G_2$, according to Definition 4 is then:

$$
ext_{edit}(G_1, G_2) = \frac{1 \times 0.18 + 0.2 \times 0.159 + 1 \times 0.22 + 0.6 \times 0.36}{1 + 0.2 + 1 + 0.6} = 0.23$$

using the weights suggested by our co-author A.C.: $w_{subn} = 1; w_{sube} = 0.2; w_{skipn} = 1; w_{skipe} = 0.6$.

3. Results

We have applied our framework, which is implemented in Java\(^3\), to stroke management processes. A stroke is the rapidly developing loss of brain function(s) due to disturbance in the blood supply to the brain. This can be due to ischemia (lack of glucose and oxygen supply) caused by a thrombosis or embolism, or to a hemorrhage. As a result, the affected area of the brain is unable to function, leading to inability to move one or more limbs on one side of the body, inability to understand or formulate speech, or inability to see one side of the visual field. A stroke is a medical emergency and can cause permanent neurological damage, complications, and death. It is the leading cause of adult disability in the United States and Europe and the number two cause of death worldwide. The best medical practice [14] requires stroke patients to be treated according to a management protocol, which is basically composed by four steps:

1. emergency management, possibly within 4h and half from symptoms onset;
2. hospitalization in a stroke unit or specialized neurological department;
3. transfer to extensive or intensive rehabilitation units when disability is still present;
4. follow-up to monitor patient compliance with drug therapy for secondary prevention.

Each step is in turn composed of a sequence of actions, which must respect some criteria, although inter-patient and inter-hospital variations are admissible. In particular, in step 1, symptom onset must be recognized, the patient must be taken to the hospital, and a brain CT must be executed. In step 2, diagnosis has to be finalized,
by means of a neurological evaluation and of several additional diagnostic investigations, meant to confirm the stroke hypothesis. Diagnostic procedures may vary, but most patients undergo an electrocardiogram and chest X-ray. Finally, a proper therapy has to be initiated: up to 90% of patients are treated with antiaggregants. Rehabilitation must also be started as soon as possible during hospitalization. In our experiments, we used traces collected from real patients, detailing the actions of steps 1 and 2. Since the care processes for ischemic and hemorrhagic stroke are very different, and the former is much more frequent, we limited our study to ischemic patients. We verified a similar patients’ cases mix in the different stroke units, so we didn’t filter the event log on the basis of particular patient parameters. Moreover, our first goal was to study the average care process, involving all the patients treated for ischemic stroke.

In particular, we asked a human expert (Dr. I. Canavero, i.e., a different person from our medical co-authors, see acknowledgments) to rate the similarity of the mined stroke management process models. We then made a comparison of our metric and of the original metric by Dijkman on these rated models. Since our metric outperformed the original one, we could confidently use it to answer a key open research question in stroke management, about the existence of a correlation between process models and clinical outcomes. Details are reported in the following sections. We could rely on a database of more than 15000 traces, collected at 35 hospitals of the Stroke Unit Network (SUN) of Regione Lombardia, Italy. Our medical co-authors belong to one of the SUN stroke units, and in particular G.M. is the SUN coordinator. Thus, they have a very deep insight into the registry data. Data refer to the period 2009–2012.

For the experiments in Section 3.1, we focused on 9929 traces, collected at the 16 stroke units with more than 250 traces. The number of traces in this subset varies from 1149 to 266. For the experiments in Section 3.2, we considered all the stroke units, and all the available traces in the SUN database. For nine stroke units, less than 200 traces were available. Traces are composed of 13 activities on average (with specific cases ranging from 3 to 90 activities).

3.1. Assessing our metric’s reliability

We asked a stroke management expert (Dr. I. Canavero) to analyze the mined process models, and, given a model as a query, order the other ones on the basis of their distance with respect to the query itself. The expert provided the ordering as a distinction among three qualitative levels, i.e., processes that have a high, medium or low similarity with respect to the query. We also ordered the available process models with respect to the same query model, resorting to: (a) the new distance defined in this paper (see Section 2.2), and (b) Dijkman’s original distance [2].

While the exercise was repeated changing the query process, for the sake of clarity in this paper we will report the results referring to one particular query model, corresponding to a single stroke unit from the SUN network. This stroke unit is a hospital in which top-level human and technological resources are available, and positive clinical outcomes (e.g., number of stroke patients who survive and/or improve after care) are very high on average. Moreover, this hospital is the one where the largest number of traces was collected (1149), and it is therefore the one from which the most reliable process mining results could be obtained. In Table 1 we report the ordering results with respect to the query model (H0), obtained by working on 15 process models, mined from the data of the other 15 stroke units from the SUN network (H1–H15).

According to the human expert (see column 2), six hospitals (H1–H6) had a high similarity, five (H7–H11) had a medium similarity, and four (H12–H15) had a low similarity with respect to the query model. Interestingly, in order to separate the hospital models into the three qualitative levels, the expert adopted a very similar rationale to the one indicated by our co-author A.C. for setting the weights (see Section 2.2.2): changes in the activities (i.e., in the graph nodes) have the highest impact, and therefore lead

<table>
<thead>
<tr>
<th>Similarity level</th>
<th>Human expert</th>
<th>New distance</th>
<th>Dijkman’s distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>H1</td>
<td>H1 (0.144)</td>
<td>H5 (0.168)</td>
</tr>
<tr>
<td>High</td>
<td>H2</td>
<td>H2 (0.144)</td>
<td>H8 (0.218)</td>
</tr>
<tr>
<td>High</td>
<td>H3</td>
<td>H9 (0.158)</td>
<td>H12 (0.232)</td>
</tr>
<tr>
<td>High</td>
<td>H4</td>
<td>H4 (0.221)</td>
<td>H6 (0.237)</td>
</tr>
<tr>
<td>High</td>
<td>H5</td>
<td>H5 (0.222)</td>
<td>H4 (0.240)</td>
</tr>
<tr>
<td>High</td>
<td>H6</td>
<td>H6 (0.235)</td>
<td>H1 (0.244)</td>
</tr>
<tr>
<td>Medium</td>
<td>H7</td>
<td>H7 (0.237)</td>
<td>H9 (0.246)</td>
</tr>
<tr>
<td>Medium</td>
<td>H8</td>
<td>H14 (0.238)</td>
<td>H15 (0.250)</td>
</tr>
<tr>
<td>Medium</td>
<td>H9</td>
<td>H3 (0.239)</td>
<td>H13 (0.253)</td>
</tr>
<tr>
<td>Medium</td>
<td>H10</td>
<td>H10 (0.240)</td>
<td>H7 (0.254)</td>
</tr>
<tr>
<td>Medium</td>
<td>H11</td>
<td>H11 (0.242)</td>
<td>H3 (0.255)</td>
</tr>
<tr>
<td>Low</td>
<td>H12</td>
<td>H12 (0.264)</td>
<td>H11 (0.256)</td>
</tr>
<tr>
<td>Low</td>
<td>H13</td>
<td>H13 (0.267)</td>
<td>H14 (0.259)</td>
</tr>
<tr>
<td>Low</td>
<td>H14</td>
<td>H8 (0.290)</td>
<td>H10 (0.263)</td>
</tr>
<tr>
<td>Low</td>
<td>H15</td>
<td>H15 (0.327)</td>
<td>H2 (0.265)</td>
</tr>
</tbody>
</table>
to the lowest similarity; changes in the activity sequence (i.e., in the graph edges), even if more penalizing than edge substitution, have a less significant impact then node substitution. This convergence of opinions between the experts somehow further supports the medical legitimacy of the choice. The distance defined in this paper (see column 3) correctly rates five of the six hospitals in the high similarity level group, three of the five hospitals in the medium similarity level group, and three of the four hospitals in the low similarity level group. Dijkman's distance (see column 4), on the other hand, correctly rates only four of the six hospitals in the high similarity level group, two of the five hospitals in the medium similarity level group, and one of the four hospitals in the low similarity level group. As already observed, in the experiments, distance weights were set according to our medical co-author A.C.'s opinion, as follows: \( w\text{sub} = 1; w\text{swh} = 0.2; w\text{kip} = 1; w\text{kipe} = 0.6 \). It is, however, worth noting that, by changing the weights and setting them to the values suggested in [9] when adopting Dijkman's distance, ordering did not change.

In conclusion, our distance appears to be more reliable, and closer to the human expert's behavior. As such, we could confidently exploit this to test the correlation between process models and clinical outcomes.

3.2. Adopting the new metric to address clinical research questions

From the clinical viewpoint, it is very interesting to verify whether structural differences in process models impact on clinical outcomes. Indeed, the frequent unavailability of human or technical resources, or, more generally, the need for adaptation to local constraints, may have led to suboptimal local practices, which may reduce the success rate of stroke care. The stroke unit we used as a query in the experiments in Section 3.1 has the highest survival rate at discharge, with respect to all the hospitals in the SUN network. We wanted to verify whether stroke units that are similar to the query also have a similar survival rate, i.e., whether the survival rate decreases as the distance from the query increases. Since the query hospital also has a very good rate of patients whose health status was improved at discharge, we wanted to address an analogous question about improvement rate. Specifically, patient improvement is defined as a reduction of the patient's score in the National Institutes of Health Stroke Scale (NIHSS). The NIHSS is a tool used by health care providers to objectively quantify the impairment caused by a stroke; 0 indicates normal functions, increasing values indicate impairments of various grades. While in the experiments in Section 3.1 we worked on 15 stroke units, in this section we extended the analysis to the models of all of the 34 stroke units from the SUN network. The human expert's ordering with respect to the query was available only for 15 of the 34 stroke units. However, since our distance proved to work at a level comparable to that of a human expert (see Section 3.1), we relied on it to produce the overall ordering, on which we then tested the correlation. Fig. 4 shows the results. As can be seen, there are lots of fluctuations. With regard to the survival rate at hospital discharge, there is a decreasing trend but it is very weak. In contrast, in terms of patient improvement from admission to discharge, the trend is much more evident, showing that the more distant a stroke unit is with respect to the query, the lower the improvement.

To explain fluctuations, we can see that the number of traces available at the different stroke units is not always the same. As observed above, some process models had to be mined relying on quite a small database (e.g., less than 200 traces, compared to the 1149 of the query hospital). In these examples, process mining results may be less reliable. Moreover, it is worth noting that there might be a stronger correlation between a part of the process and the clinical outcomes: specifically, emergency management could have a deeper impact on care outcomes. The clinical relevance of this method resides also in the fact that, after showing that different processes lead to different outcomes, stroke managers are fostered to deeply analyze the processes themselves to point out weaknesses and find solutions. In particular, they could focus on the processes that lack some of the actions present in the reference process, to find out which ones are more correlated to the outcome. An example of this investigation can be found in [15].

As a further assessment of the reliability of our metric, we have also verified that the evident decreasing trend in patient improvement observed in Fig. 4 cannot be found when adopting Dijkman's metric. Indeed, Fig. 5 compares the values and the trend behavior of patient improvement, when stroke units are ordered according to our metric (top), and according to Dijkman's metric (bottom), with respect to their distance from the query hospital H0. The comparison involves 15 stroke units (as explained in Section 3.1, Dijkman's distance values are available only for 15 hospitals. The figure plots the data in Table 1). As it can be observed, when adopting our metric, the more the process model of a stroke unit is distant from the process model of H0, the lower is its average patient improvement. On the other hand, this kind of trend cannot be observed when relying on Dijkman's distance (interestingly, the trend line is almost flat in the bottom part of the figure).

In the future, we plan to repeat these tests, after having collected more traces at the various hospitals, and to repeat the experiments by focusing on subparts of the process models, in order to address the limitations outlined above. Moreover, we would like to analyze additional performance indicators, such as the number of surviving and/or improved patients, after a follow-up period. These data are harder to obtain, because they are not directly available at the stroke unit, but must be gathered by directly contacting the patients (who are not hospitalized anymore). Therefore, the completion of this experimental task will require some more time. However, we believe that these further investigations will help medical experts to gain a better understanding of the quality of stroke care at the various units from the SUN.

4. Comparison to related works

Graph representation and retrieval is a very active research area, which is giving birth to different methodological approaches and software tools. Graph databases, like, e.g., HypergraphDB [16] and DEX [17], are gaining popularity, for working in emerging linked data such as social network data and biological data. However, in this section we will focus on contributions that are more closely related to comparison and retrieval in process/workflow management research. As clearly stated in [9,13], three classes of similarity metrics can be considered to deal with process model comparisons: (i) node (or edge) matching similarity, which basically compares the labels attached to process model nodes (or edges); (ii) structural similarity, which compares node labels, as well as graph topology; (iii) behavioral similarity, which compares node labels, as well as the behavioral/causal relations captured in the process models. While class (i) somehow oversimplifies the problem, class (iii) requires causal information, which we do not currently mine. Indeed, our work is related to class (ii). Therefore, in Section 4.1, we will focus on structural similarity approaches. In Section 4.2, we will comment on techniques to contain computational complexity. Finally, in Section 4.3 we will report on works combining process mining with process retrieval and comparison, and on medical applications.
4.1. Structural similarity approaches for process model comparison

The goal of comparing objects with a complex structure (i.e., graphs) entails the definition of a nontrivial notion of distance. The issue of providing a proper graph distance definition has been afforded in the literature, following three main directions, i.e.:

1. relying on a local notion of similarity (two subgraphs are similar if their neighboring nodes are similar), as in the similarity flooding algorithm [18];
2. relying on subgraph isomorphism, e.g., to find maximum common sub-graphs [19], and
3. adapting the edit distance notion to graphs [3].

We are currently following direction (3), but directions (1) and (2) could be considered in our future work for comparison.

The work in [20] describes an approach related to direction (1). In [20] a retrieval system for supporting incremental workflow modeling is presented. The system proposes a similarity-based reuse of workflow templates using a planner that employs an inexact graph matching algorithm based on similarity flooding. For computing similarities, the algorithm relies on the idea that elements of two distinct graphs are similar, when their adjacent elements are similar. The algorithm propagates the similarity from a node to its respective neighbors based on the topology in the two graphs. However, edge similarity is not considered. The work in [21] belongs to direction (2), as it exploits a maximum common subgraph approach for process retrieval, in a retrieval system for supporting business process monitoring. Interestingly, the metric in [21] takes into account temporal information, since it combines a contribution related to activity similarity, and a contribution related to delays between activities. The approach in [22] relies on graph isomorphism (direction (2)) for retrieving scientific workflows (e.g., pipelines for bioinformatics experiments). Unlike business workflows, scientific workflows have a strong focus on the data flow, typically restricting the control flow to a partial ordering of the tasks. The peculiarity of scientific workflows makes this contribution less closely related to our work. The works following direction (3), on the other hand, extend the notion of graph edit distance [3], which calculates the minimal cost of transforming one graph into another, by applying insertions/deletions and substitutions of nodes, and insertions/deletions of edges. Very interestingly, all the structural similarity approaches discussed in the survey in [13] adopt direction (3). Some of them are discussed below.

The work in [2], which we have extended, provides a normalized version of the graph edit distance [3] for comparing business process models, and defines syntactical edit operation costs, which we have partially reused, as discussed in Section 2.2. The authors do not take into account control flow elements other than sequence. Very similarly, the work in [23] also relies on graph edit distance, and exploits string edit distance on node names to determine the cost of node substitutions. The work in [24] encapsulates a set of edit operations into the so-called “high-level change operations”, and measures distance on the basis of the number of high-level change operations needed to transform one graph into another. The work in [25] transforms a graph into an ordered tree, and then exploits tree edit distance. The work in [26] also makes use of a normalized version of the graph edit distance. The approach is used to support workflow modification in an agile workflow system, and takes into account control flow information as well as activity information. In particular, control flow elements other than sequence (i.e. AND/OR splits and joins) are explicitly modeled as nodes. Enhancing our distance definition in order to deal with these kinds of nodes is one of our future research directions, as discussed in Section 5. However, [26] only makes use of syntactical information in the definition of the edit operation costs. Moreover, the work is limited to considering (small) changes with respect to a running process instance. With respect to [2,23–26], we make use of semantic information too, by relying on taxonomic distance in the comparison.
between mapped (i.e., substituted) nodes. We also make explicit use of the information mined/learned from the data in the mapped edges contribution.

The use of semantic information in structured process model retrieval is proposed in [27], a system working on workflows represented as semantically labeled graphs. The work in [27] adopts a graph edit distance-based approach, which is particularly suitable for scientific workflows. The paper proposes using a metric in which the similarity between two mapped nodes or edges makes explicit use of their semantic description. However, the framework is presented in a general, high-level way, and the specific costs of edit operations are not provided. The approach adopts the so-called local-global principle [28], in which a local distance function is defined for each individual property of the model (e.g., the mapped nodes), reflecting the goodness of fit of a model with respect to that single property only. The local distances are then aggregated into the global distance by means of an aggregation function. This function appropriately combines all local distance values (e.g., by a weighted average) into a single value that aims at reflecting the goodness of fit of the model as a whole. Many academic and commercial retrieval tools relying on the case-based reasoning methodology [29] (e.g., JColibri [30]) use this approach.

We have adopted the local-global principle too. With respect to our work, however, [27] is much more focused on the data flow, which was not considered in our current application. As already observed, this makes this work less related to ours.

Finally, the SAI toolkit [31] is transversal with respect to the three directions (1), (2) and (3) outlined above, since it is a framework for workflow case representation and retrieval that allows different domain-specific similarity measures to be used.

4.2. Dealing with computational complexity

Most of the approaches following directions (1), (2) and (3) typically suffer from problems related to their high computational complexity, which is sometimes mitigated by resorting to greedy techniques (see, e.g., [2] and our own approach). To avoid these problems, however, some previous works have investigated structureless workflow retrieval, where workflow representation is a plain textual description, a set of tags [22], or a set of abstract workflow features [32]. Other approaches have suggested the use of a two-step procedure, which combines an initial and comparatively inexpensive retrieval step, to winnow the structured cases to be considered, with a more expensive strategy that ranks the

**Fig. 5.** Plots showing patient improvement rate (in %) of the SUN stroke units depending on their distance from the query model. On the x-axis: ordering of the stroke units according to their distance from the query, by exploiting our metric (top), and Dijkman’s metric (bottom). Linear fitting is shown as well.
remaining cases (as in the well-known MAC/FAC system [33]) [31,34]. In [27] some procedures for non-exhaustive search, based on the A* algorithm, are provided. In [35], preference-based case-based reasoning is used to enhance retrieval in the CavBase database [36], a database of protein structure that exploits maximum common subgraph calculation for retrieval. In preference-based case-based reasoning, problem-solving experience is mainly represented in the form of preferences for candidate solutions in the context of a target problem to be solved. This technique is suitable when the evaluation of candidate solutions is expensive, so that only a relatively few candidates can be tried in a problem-solving episode before a selection is made. More work on computational performances may be considered in our future research as well.

4.3. Process mining and healthcare applications

A few approaches combining mining techniques with retrieval are reported in the literature. Among them it is worth citing the Phala system [37], which supports the generation and composition of scientific workflows by mining execution traces for recommendations to aid workflow authors. The system relies on unnormalized graph edit distance. In particular, to generate cases, Phala mines the data flow between services within an instance of workflow execution. Therefore, the focus of this work is on the data flow, and not on the process (control flow), which is less relevant in scientific workflows. As already observed for the works in [27,22], this makes the contribution less related to our work.

Process mining in healthcare is a field that is gaining attention in recent years. In [38] process mining in the gynecological oncology domain has been considered; in [39] several processes within an emergency department have been investigated; in [40] imaging diagnostics procedure appointments within a radiology workflow have been analyzed; in [41] process mining has been applied to stroke data; in [42] the activities that are performed during hospitalization for breast cancer are investigated; in [43] the journey through multiple wards has been discovered for inpatients; in [44] the workflow of a laparoscopic surgery has been analyzed. The contribution in [38] also proposes pre-processing techniques to improve process mining (which are highly recommended in the medical domain, due to its complexity, as discussed in [45]). All of these systems, however, do not deal with process retrieval and comparison. The work in [46] is an overview about challenges and issues of process mining in healthcare, especially referring to the fact that needed data may be spread across different data sources, that may rely on different granularities in timestamp recording. Other works dealing with process management and analysis in healthcare include, e.g., [47], a system able to support adaptations of running healthcare process instances, and [48,49], both dealing with conformance checking techniques of clinical guidelines and pathways. In summary, the panorama on process mining and management in healthcare, the domain to which we are applying our framework, is huge and very active. However, it is worth noting that our work is absolutely general. Indeed, we plan to test it in nonmedical domains as well.

• in distance calculation, we currently simplify the control flow information of the mined models, by simply considering sequence, and ignoring AND/OR splits and joins. In the future, we plan to remove this limitation, by explicitly representing this information through gateway nodes. This enhancement in the process models representation will require a post-processing step after the generation of the Heuristic Miner output. Once available, this information will be used to better characterize the comparison between two graphs: as in [26], we will distinguish between activity nodes and gateway nodes, and separately manage them in the mapping process. With respect to [26], however, we will adopt a semantical (and not just syntactical) approach in the comparison of activity nodes (as we already do in the current version of the work). This extension will also represent a significant improvement with respect to [2], where control flow information other than sequence is disregarded;
• we currently disregard one output provided by Heuristic Miner, namely model fitness. This metric quantifies how well the event log can be replayed in the mined model. Comparing two graphs with very different fitness values may lead to a biased result, if fitness is ignored. In our experiments, we could observe that fitness values were usually relatively similar in practice (they ranged from 0.34 to 0.44, being fitness defined as $e_{[0,1]}$). However, this may be not the case in other situations. In order to enhance the generality of our approach, and its applicability to other domains, or to situations in which available data are of different quality, we plan to modify the distance definition, in order to properly take model fitness into account;
• from a more technological point of view, at the moment, we have implemented our approach as a stand-alone Java tool, from which Heuristic Miner has to be explicitly invoked. In the future, we plan to integrate our distance calculation as a plug-in in the ProM 6 environment. This will allow us to make our work available to the process mining community, facilitating the collection of feedback from other users, and the testing also in very different application domains;
• the distance definition we propose has been tailored to the output of the Heuristic Miner algorithm, which is one of the most “popular” miners in ProM, but surely not the only one. Indeed, while Heuristic Miner can be particularly suited for some applications (e.g., because of its tolerance to noise), other algorithms may be fruitfully adopted in different situations. In the future, we would thus like to test different process mining algorithms as well, to increase the practical applicability of our approach. The exploitation of a different algorithm will impact on distance calculation: adjustments may be required, in order to take into account specific outputs that were not provided by Heuristic Miner. We believe that the integration of our work in the form of a ProM plug-in, as discussed above, will however facilitate this last step.

6. Conclusions

This work showed that process mining and process comparison can be applied successfully to clinical data to gain a better understanding of medical processes. It is interesting to analyze the differences, to establish whether they concern only the scheduling of the various tasks or also the tasks themselves. In this way, not only may different practices that are used to treat similar patients be discovered, but also unexpected behavior may be highlighted. Experimental results have shown the quality of our contribution, in comparison to the distance definition reported in [2]. Indeed, our metric outperformed Dijkman’s [2], and provided results that were closer to those of a human expert. Thus, we could confidently use our metric to answer a very interesting research question in
stroke management, about the existence of a correlation between process models and survival and improvement rate at discharge. Despite the presence of fluctuations, which can be explained by the relatively low number of traces available at some stroke units, a decreasing trend could be identified, suggesting that the greater the distance from the reference hospital, the worse these clinical outcomes become. In the future, we would like to test the correlation with additional clinical outcomes as well, and we will repeat the experiment by focusing on subparts of the process model (e.g., emergency management), which could be particularly critical in determining patient improvement. We would also like to experiment the framework in the myocardial infarction domain. Indeed, myocardial infarction regional and national registry data are available to us, on which process mining and process comparison could be applied. From the methodological and technological viewpoint, we will extend the approach along the research lines discussed in Section 5. We believe that these enhancements can represent a relevant added value to our work, by making process comparison more versatile, reliable and useful in practice. Further experiments, possibly in different application domains, will be designed in order to support this claim.

Acknowledgments

This research is partially supported by the GINSENG Project, Compagnia di San Paolo. We would like to thank Dr. I. Canavero for the independent evaluation of process distance.

References


