Visualization and Clustering of Business Process Collections Based on Process Metric Values

Joachim Melcher and Detlef Seese
Institut AIFB, Universität Karlsruhe (TH)
76128 Karlsruhe, Germany
{melcher|seese}@aifb.uni-karlsruhe.de

November 17, 2008

Abstract

Motivated by ideas of software measurement, the area of process measurement has attracted attention in recent time. Numerous process metrics have been proposed to measure (often structural) properties of business processes. These metric values can be used to characterize and compare processes. Integrated into valid prediction systems, they can be useful for predicting external process attributes like duration, costs, number of errors or understandability. As this area of research is quite young, not much knowledge about the behavior of these metrics (e. g., distribution of metric values and correlations between metrics) exists.

In this paper, we propose heatmaps, a visualization technique for high-dimensional data originally used in genetics, for visualizing the process metric values of business process collections. So, new insights into the distribution of the metric values among the processes could be gained. Additionally, we use clustering for analyzing (1) the correlations between different process metrics and (2) finding (structurally) similar processes among business process collections. Our approach has been successfully applied to the SAP Reference Model processes.

1 Introduction

During the previous decades, the field of software measurement has created theoretical concepts for measuring software and making predictions on software quality attributes (see, e. g., [4] for an overview). Motivated by this research, several papers proposing process metrics have been published in recent years. These metrics measure (often structural) properties of business processes and can be used to characterize and compare processes. Integrated into valid prediction systems, they can be useful for predicting external process attributes like duration, costs, number of errors or understandability. As this area of research is quite young, not much knowledge about the behavior of these metrics (e. g., distribution of metric values and correlations between metrics) exists.

To gain new insights into these questions, the visualization and analysis of the process metric values of large business process collections would be interesting. The resulting process metric data would be very high-dimensional making visualization problematic.

In this paper, we propose heatmaps, a visualization technique for high-dimensional data originally used in genetics, for visualizing the process metric values of business process collections. So, new insights into the distribution of the metric values among the processes could be gained.

Additionally, we use clustering for analyzing (1) the correlations between different process metrics and (2) finding (structurally) similar processes among business process collections. The clustering does not consider *behavioral* similarity as, for example, in [13].

Finally, we apply our approach to the SAP Reference Model processes.

The remainder of this paper is organized as follows: In Section 2, we give a short overview about the area of process measurement. The use of heatmaps for visualizing the high-dimensional process metric data of business process collections is explained in Section 3. In Section 4, we present basics on clustering. The results of an experimental application of our approach are given in Section 5. The paper gives a conclusion and presents possible future work (Section 6).

2 Process Measurement

The area of process measurement is inspired by the works and results of software measurement. Several papers proposing process metrics have been published in recent years (see [7, pp. 1–2] for an overview).

According to Fenton and Pfleeger, there are two main types of measurement:

Definition 1 (Measurement systems) *Measurement systems are used to assess an* existing *entity by numerically characterizing one or more of its attributes [4, p. 104].*

Definition 2 (Prediction systems) *Prediction systems are used to predict some attribute of a* future *entity, involving a mathematical model with associated prediction procedures [4, p. 104].*

Besides the use for *future* entities, as stated in the definition of Fenton and Pfleeger, prediction systems can also be used to predict some attribute of an *existing* entity that is measurable only in a very laborious manner.

In [7], we show how the idea of prediction systems can be transfered to process measurement (see Figure 1):

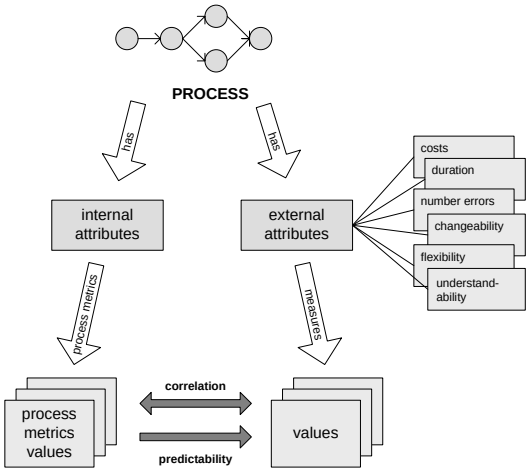

Figure 1: Prediction systems adapted to process measurement.

A process has *internal* and *external* attributes.

Internal attributes can be measured purely in terms of the process separate from its behavior [4, p. 74]. Most proposed process metrics measure structural properties (internal attributes).

External attributes can be measured only with respect to how the process relates to its environment [4, p. 74]. Examples are costs, duration, number of errors and understandability.

3 Heatmaps

The process metric data of (large) business process collections is high-dimensional data with many data vectors. So, the problem arises how to visualize this data.

Several existing methods are available, but all of them have big disadvantages:

- Scatter plots (see Figure 2 for an example) are good for visualizing large amounts of data vectors. But they are only applicable for 2D or at most 3D data.

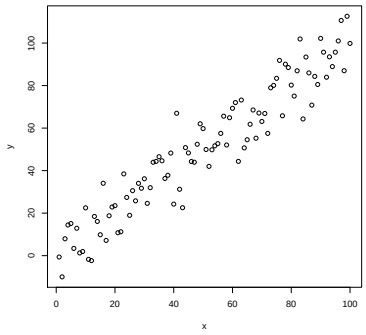

Figure 2: Example of a scatter plot.

- Radar charts[1] (see Figure 3 for an example) are drawn in two dimensions and can display data with three or more dimensions. For each dimension, there exists an axis. The axes start in one single center point and are uniformly placed around the 360° of a circle. The points on these axes form a polygon representing one vector. Radar charts soon become confusing when increasing the number of dimensions and depicting many data vectors.

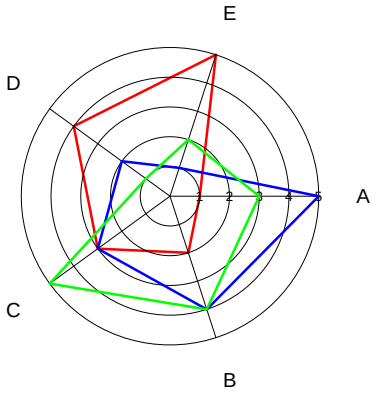

Figure 3: Example of a radar chart.

[1] Radar charts are also called spider charts or star charts.

To overcome these problems, we propose the use of heatmaps, a visualization technique originally used in genetics for depicting microarray data. Recently, this method was adapted to visualizing the individuals (i. e., possible solutions) of population based multi-objective algorithms (e. g., genetic algorithms) [12].

A heatmap displays the data as a matrix: one row per data vector and one column per dimension (see Figure 5 for an example). The values of the cells are color-coded—blue for minimum values and red for maximum values (see Figure 4). The different dimensions can be individually normalized into the interval $[0, 1]$ if their domains are too different.

Figure 4: Color legend for heatmaps (blue for minimum and red for maximum values).

Heatmaps have many advantages compared to other visualization methods for high-dimensional data: Large amounts of data can be clearly displayed on one page. Correlations between different dimensions and the distribution of the values of the different dimensions become visible.

For our case, the process metric values of a process are displayed in one row. The different process metrics form the columns of the matrix. External attributes (as duration, costs, number of errors or understandability) can be added as additional columns of the heatmap if desired.

4 Clustering

4.1 Basics

A good overview about clustering is given by Berkhin in [1].

The general goal of clustering is to partition a set $X \subseteq \mathbb{R}^n$ of data points into k subsets (clusters) $\mathcal{C} = \{C_1, \ldots, C_k\}$. These clusters are disjoint—their union is equal to the full set of data points.

$$X = C_1 \cup C_2 \cup \cdots \cup C_k, \quad C_i \cap C_j = \emptyset,\ i \neq j \quad (1)$$

Two often used methods in practice are hierarchical and partitive clustering. These are explained in more detail in the following subsections.

4.2 Hierarchical Clustering

The result of a hierarchical clustering is a so called clustering tree (dendrogram) (see the top of Figure 5 for an example). Each node of this tree has a corresponding cluster. The corresponding cluster of a node is the union of all clusters belonging to this node's child nodes.

Hierarchical clustering can be divided into agglomerative (bottom-up) and divisive (top-down) algorithms for constructing the clustering tree.

In this paper, agglomerative hierarchical clustering is used. The approach is described in pseudo code in Algorithm 1.

Algorithm 1 Agglomerative hierarchical clustering.

Function AGGLHIERARCHICALCLUSTERING(X)

Input: set X of data vectors

Output: clustering tree (dendrogram) D

```
1:  C ← ∅
2:  for i = 1 to |X| do
3:      {initialize: assign each vector to its own cluster}
4:      C_i ← {x_i}
5:      C ← C ∪ {C_i}
6:  end for
7:  numberClusters ← |X|
8:  repeat
9:      {compute distances between all clusters}
10:     for all C_i ∈ C do
11:         for all C_j ∈ C do
12:             if C_i ≠ C_j then
13:                 compute distance d(C_i, C_j) between clusters C_i
                    and C_j
14:             end if
15:         end for
16:     end for
17:     {merge the two clusters C_i and C_j that are closest to each
        other}
18:     C_{i,j} ← C_i ∪ C_j
19:     C ← C\{C_i, C_j} ∪ {C_{i,j}}
20:     numberClusters ← numberClusters − 1
21:     {store information about two sub-clusters}
22:     C_{i,j}.child1 ← C_i
23:     C_{i,j}.child2 ← C_j
24:     D ← C_{i,j}
25: until numberClusters = 1
26: return D
```

For the inter-cluster distance $d(C_i, C_j)$ in line 13 of Algorithm 1, several measures exist. Among these are

- single linkage

$$d_s(C_i, C_j) := \min_{\substack{\vec{x}_i \in C_i \\ \vec{x}_j \in C_j}} \{d(\vec{x}_i, \vec{x}_j)\} \quad , \quad (2)$$

- complete linkage

$$d_{co}(C_i, C_j) := \max_{\substack{\vec{x}_i \in C_i \\ \vec{x}_j \in C_j}} \{d(\vec{x}_i, \vec{x}_j)\} \quad \text{and} \quad (3)$$

- average linkage

$$d_a(C_i, C_j) := \frac{1}{|C_i||C_j|} \sum_{\substack{\vec{x}_i \in C_i \\ \vec{x}_j \in C_j}} d(\vec{x}_i, \vec{x}_j) \quad . \quad (4)$$

In each of these measures, $d(\vec{x}_i, \vec{x}_j)$ is a distance measure between the two vectors $\vec{x}_i$ and $\vec{x}_j$. This could be, for example, the Euclidean distance $\|\vec{x}\|_2$ of (5).

$$\|\vec{x}\|_2 := \sqrt{\sum_{i=1}^{n} |x_i|^2} \quad , \vec{x} \in \mathbb{R}^n \tag{5}$$

4.3 Partitive Clustering: k-means

The k-means clustering algorithm is a randomized clustering approach that generates a disjoint, non-hierarchical partitioning consisting of k clusters. The algorithm is described in pseudo code in Algorithm 2.

Algorithm 2 k-means clustering.

Function KMEANS(X, k)

Input: set X of data vectors, number of clusters k

Output: clustering $\mathcal{C}$ with k clusters

```
1:  C ← ∅
2:  for i = 1 to k do
3:      C_i ← ∅
4:      C ← C ∪ {C_i}
5:      randomly initialize cluster center (centroid) c_i
6:  end for
7:  repeat
8:      {compute partitioning for data}
9:      for i = 1 to k do
10:         C_i ← ∅
11:     end for
12:     for j = 1 to |X| do
13:         add x_j to that C_i with shortest Euclidean distance between x_j and c_i
14:     end for
15:     {update cluster centers}
16:     for i = 1 to k do
17:         c_i := 1/|C_i| Σ_{x_j ∈ C_i} x_j
18:     end for
19: until partitioning stays unchanged or the algorithm has converged
20: return C
```

It minimizes the error $E(\mathcal{C})$ with

$$E(\mathcal{C}) = \sum_{i=1}^{k} \sum_{\vec{x_j} \in C_i} \|\vec{x_j} - \vec{c_i}\|_2^2 \quad . \tag{6}$$

As the k-means algorithm does not depend on previously found sub-clusters, it often results in better clusterings than gained with hierarchical approaches. Yet, as it is a randomized algorithm, its execution is indeterministic—possibly resulting in several different clusterings for the same data set X and value k. So, the question arises how to choose the number k of clusters and how to choose from the different clusterings potentially found for the same number of clusters.

One possible solution to this problem is the Davies-Bouldin index [3] defined as

$$DB(\mathcal{C}) := \frac{1}{k} \sum_{i=1}^{k} \max_{\substack{j \in \{1,\ldots,k\} \\ i \neq j}} \left\{ \frac{S_c(C_i) + S_c(C_j)}{d_{ce}(C_i, C_j)} \right\} \quad . \tag{7}$$

Thereby, S_c is defined as

$$S_c(C_i) := \frac{1}{|C_i|} \sum_{\vec{x}_j \in C_i} \|\vec{x}_j - \vec{c}_i\|_2 \tag{8}$$

and acts as a dispersion measure quantifying the average centroid distance of the cluster's vectors.

The measure d_{ce} is defined as

$$d_{ce}(C_i, C_j) := \|\vec{c}_i - \vec{c}_j\|_2 \tag{9}$$

and quantifies the distance between two clusters (centroid linkage).

An optimal clustering consists of "compact" clusters with small dispersion and large distances between the single clusters. Looking at (7), one can easily notice that such an optimal clustering minimizes the value of the Davies-Bouldin index.

5 Experimental Application of Approach

5.1 Selected Process Metrics

As already stated, numerous process metrics are proposed in the literature. Yet, they require different process representations (e. g., Petri nets, workflow nets or EPCs). In order to compare the process metrics, we had to choose metrics that are applicable for the same process representation. Looking at a recent overview about proposed process metrics [7, pp. 1–2], we chose metrics for EPCs.

A business process model (in EPC representation) is a special kind of graph $G = (N, A)$ consisting of a set N of nodes and a set $A \subseteq N \times N$ of arcs. There are two node types: tasks T and connectors C ($N = T \cup C$). Tasks can be functions F or events E ($T = F \cup E$), connectors can be splits S or joins J ($C = S \cup J$). Each connector has one of the labels AND, XOR or OR. Each connector $c \in C$ has an in-degree $d_{in}(c) = |\{(n_1, n_2) \in A | n_2 = c\}|$, an out-degree $d_{out}(c) = |\{(n_1, n_2) \in A | n_1 = c\}|$ and a degree $d(c) = d_{in}(c) + d_{out}(c)$.

The 33 selected EPC process metrics are listed in Table 1.

5.2 Selected Processes

We selected the SAP Reference Model [2,5], which was part of SAP R/3 until version 4.6, as process collection

Table 1: Selected process metrics for EPCs.

name	symbol	reference	definition
number start events	S_{E_S}	[8, 10]	
number internal events	$S_{E_{Int}}$	[8, 10]	
number end events	S_{E_E}	[8, 10]	
number events	S_E	[10]	$S_E(G) = \|E\| = S_{E_S}(G) + S_{E_{Int}}(G) + S_{E_E}(G)$
number functions	S_F	[8, 10]	$S_F(G) = \|F\|$
number AND splits	$S_{S_{AND}}$	[8, 10]	
number AND joins	$S_{J_{AND}}$	[8, 10]	
number XOR splits	$S_{S_{XOR}}$	[8, 10]	
number XOR joins	$S_{J_{XOR}}$	[8, 10]	
number OR splits	$S_{S_{OR}}$	[8, 10]	
number OR joins	$S_{J_{OR}}$	[8, 10]	
number connectors	S_C	[10]	$S_C(G) = \|C\| = S_{S_{AND}}(G) + S_{J_{AND}}(G) + S_{S_{XOR}}(G) + S_{J_{XOR}}(G) + S_{S_{OR}}(G) + S_{J_{OR}}(G)$
number nodes	S_N	[10]	$S_N(G) = \|N\| = S_E(G) + S_F(G) + S_C(G)$
number arcs	S_A	[8, 10]	$S_A(G) = \|A\|$
diameter	$diam$	[10]	length of the *longest* path (= number of arcs on this path) from a start node to an end node
density (1)	Δ	[10]	$\Delta(G) = \frac{\|A\|}{\|N\| \cdot (\|N\|-1)}$: number of arcs divided by the maximum number of arcs for the same number of nodes
density (2)	d	[9]	see [9, pp. 3–4]
coefficient of connectivity	CNC	[6, 10]	$CNC(G) = \frac{\|A\|}{\|N\|}$
coefficient of network complexity	CNC_K	[6]	$CNC_K(G) = \frac{\|A\|^2}{\|N\|}$
cyclomatic number	S	[6]	$S = \|A\| - \|N\| + 1$
avg. connector degree	$\overline{d_C}$	[10]	$\overline{d_C}(G) = \frac{1}{\|C\|} \sum_{c \in C} d(c)$ (see [a])
max. connector degree	$\widehat{d_C}$	[10]	$\widehat{d_C}(G) = \max\{d(c) \| c \in C\}$ (see [a])
separability	Π	[10]	$\Pi(G) = \frac{\|\{n \in N \| n \text{ is cut-vertex}\}\|}{\|N\|-2}$: A cut-vertex is a node whose deletion separates the process model into multiple components.
sequentiality	Ξ	[10]	$\Xi(G) = \frac{\|A \cap (T \times T)\|}{\|A\|}$: number of arcs between non-connector nodes divided by the number of arcs
depth	Λ	[10]	Depth relates to the maximum nesting of structured blocks in a process. See [10, pp. 185–186].
mismatch	MM	[10]	$MM(G) = \sum_{l \in \{AND,XOR,OR\}} \left(\left\| \sum_{c \in S_l} d_{out}(c) - \sum_{c \in J_l} d_{in}(c) \right\| \right)$ (see [b]): sum of mismatches for each connector type
heterogeneity	CH	[10]	$CH(G) = -\sum_{l \in \{AND,XOR,OR\}} p(l) \cdot \log_3 p(l)$ (see [a]): entropy over the different connector types
cycling	CYC	[10]	$CYC_N(G) = \frac{\|N_C\|}{\|N\|}$: number of nodes N_C on a cycle divided by the number of nodes
token splits	TS	[10]	$TS(G) = \sum_{c \in S_{AND} \cup S_{OR}} (d_{out}(c) - 1)$: number of newly introduced tokens by split connectors
control flow complexity	CFC	[8, 10]	$CFC(G) = \sum_{c \in S_{AND}} 1 + \sum_{c \in S_{XOR}} d_{out}(c) + \sum_{c \in S_{OR}} \left(2^{d_{out}(c)} - 1\right)$: sum over all split connectors weighted by their number of possible states after the split
join complexity	JC	[8]	$JC(G) = \sum_{c \in J_{AND}} 1 + \sum_{c \in J_{XOR}} d_{in}(c) + \sum_{c \in J_{OR}} \left(2^{d_{in}(c)} - 1\right)$: sum over all join connectors weighted by their number of possible states before the join
weighted coupling metric	CP	[14]	see [14, p. 42]
cross-connectivity metric	CC	[15]	average strength of connection between all pairs of process nodes, see [15, pp. 483–484] for details

[a] Metric value is 0 for $|C| = 0$ (source: personal communication with Jan Mendling).
[b] The original definition printed in [10, p. 187] is faulty (source: personal communication with Jan Mendling).

for our experiment. These processes were already used for several experiments found in the literature [8–10].

We first validated the EPCs according to the requirements for *syntactically correct EPCs* [10, pp. 42–46]. Furthermore, we discarded EPCs with several graph components. Out of the 604 non-trivial EPCs of the SAP Reference Model, we had to remove 89 because of invalidity[2] or several graph components.

Finally, 515 EPCs remained for the following experiment with our approach.

[2] no start event, no end event, a function with not exactly one predecessor and one successor node or an event with more than one predecessor or successor node

5.3 Results

The 33 process metric values of the 515 selected processes are depicted in the heatmap of Figure 5.

The values of each process metric are normalized into the interval $[0, 1]$ as their domains are too different. The metrics control-flow complexity (CFC) and join complexity (JC) are logarithmically normalized as both have some outliers with extremely high values compared to the large rest of the values.

The rows (i. e., processes) are ordered by the number of nodes metric (S_N). The columns (i. e., process metrics) are hierarchically clustered using $1 -$ the Spearman's rank correlation coefficient [11, pp. 42–45] as distance between two columns (process metrics) within the complete linkage inter-cluster distance measure of equation (3).

The data is clearly displayed in the heatmap on one page. So, the main goal of the visualization is fulfilled. Furthermore, several observations can be made:

- There is a strong positive correlation between the size metrics number of connectors (S_C), number of events (S_E), number of nodes (S_N) and number of arcs (S_A).
- There is a negative correlation between most metrics (e. g., size metrics) and the metrics separability (Π), sequentiality (Ξ), cross-connectivity (CC), density (1) (Δ) and weighted coupling (CP). The negative correlation is especially strong between S_C, S_E, S_N and S_A on the one side and Δ and CP on the other.
- Most metrics have many small and only some large values. For heterogeneity (CH), things are vice versa. For the metrics separability (Π) and coefficient of connectivity (CNC), most processes have values in the middle of the domain.

A clustered version of the heatmap is depicted in Figure 6. The clustering was done using the k-means clustering algorithm for three clusters. Before clustering, the input data (normalized metric values from the non-clustered heatmap) was scaled to mean 0 and variance 1 for each dimension. The selection of the optimal number of clusters and the optimal clustering with this cluster number for the input data was done using the Davies-Bouldin index.

6 Conclusion and Future Work

In this paper, we proposed heatmaps as a visualization technique for the high-dimensional process metric data of business process collections to gain new insights into the distribution of metric values among processes. Additionally, we suggested clustering for analyzing the correlations between process metrics and finding (structurally) similar processes among business process collections.

We successfully applied our approach to the SAP Reference Model processes. We could demonstrate that the visualization of 33 process metric values for 515 processes using heatmaps is possible and still clear for a human observer. Furthermore, interesting insights into the correlations between process metrics and the clustering of the processes of the collection could be gained.

For future work in this area, we suggest to apply the approach also to other process collections. It would be interesting to analyze whether these processes have similar correlations between the process metrics and a similar distribution of metric values as the processes examined in this paper.

Acknowledgment

The authors want to thank Sanaz Mostaghim for her fruitful remarks on heatmaps.

References

[1] P. Berkhin. A survey of clustering data mining techniques. In Jacob Kogan, Charles Nicholas, and Marc Teboulle, editors, *Grouping Multidimensional Data: Recent Advances in Clustering*, pages 25–71. Springer, 2006.

[2] Thomas Curran, Gerhard Keller, and Andrew Ladd. *SAP R/3 Business Blueprint : Understand the Business Process Reference Model*. Prentice Hall, 1997.

[3] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[4] Norman E. Fenton and Shari Lawrence Pfleeger. *Software Metrics: A Rigorous and Practical Approach*. International Thomson Computer Press, 2nd edition, 1996.

[5] Gerhard Keller and Thomas Teufel. *SAP R/3 Process-oriented Implementation : Iterative Process Prototyping*. Addison Wesley, 1998.

[6] Antti M. Latva-Koivisto. Finding a complexity measure for business process models. Research report, Helsinki University of Technology, Systems Analysis Laboratory, 2001.

[7] Joachim Melcher and Detlef Seese. Process measurement: Insights from software measurement on measuring process complexity,

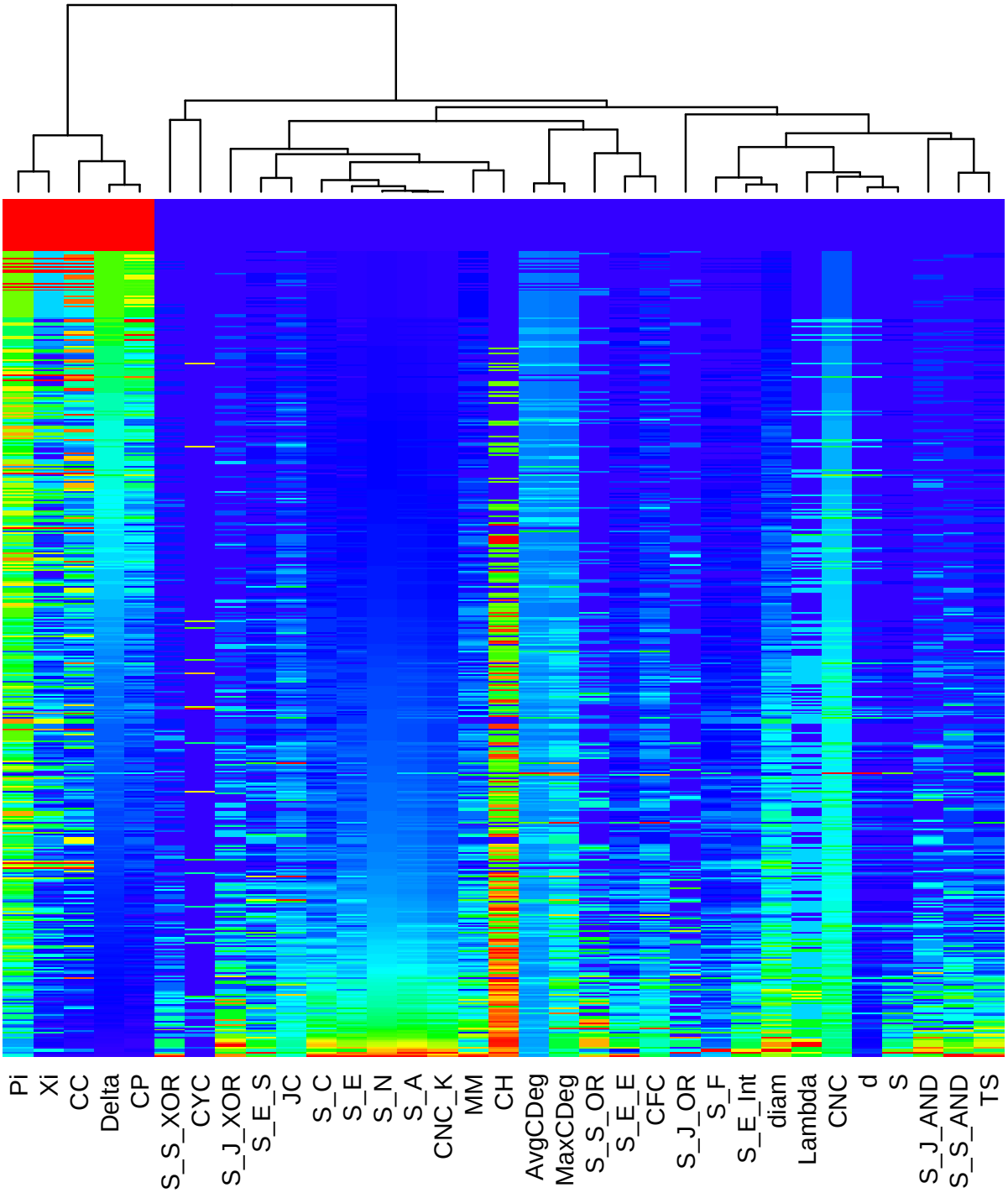

Figure 5: Heatmap displaying 33 process metric values for 515 processes. The rows are ordered by the number of nodes (S_N). The columns are hierarchically clustered using $1 -$ the Spearman's rank correlation coefficient as distance measure.

quality and performance. Research report, Universität Karlsruhe (TH), Institut AIFB, 2008. http://digbib.ubka.uni-karlsruhe.de/volltexte/1000009225.

[8] J. Mendling, M. Moser, G. Neumann, H. M. W. Verbeek, B. F. van Dongen, and W. M. P. van der Aalst. A quantitative analysis of faulty EPCs in the SAP reference model. BPM Center Report BPM-06-08, 2006.

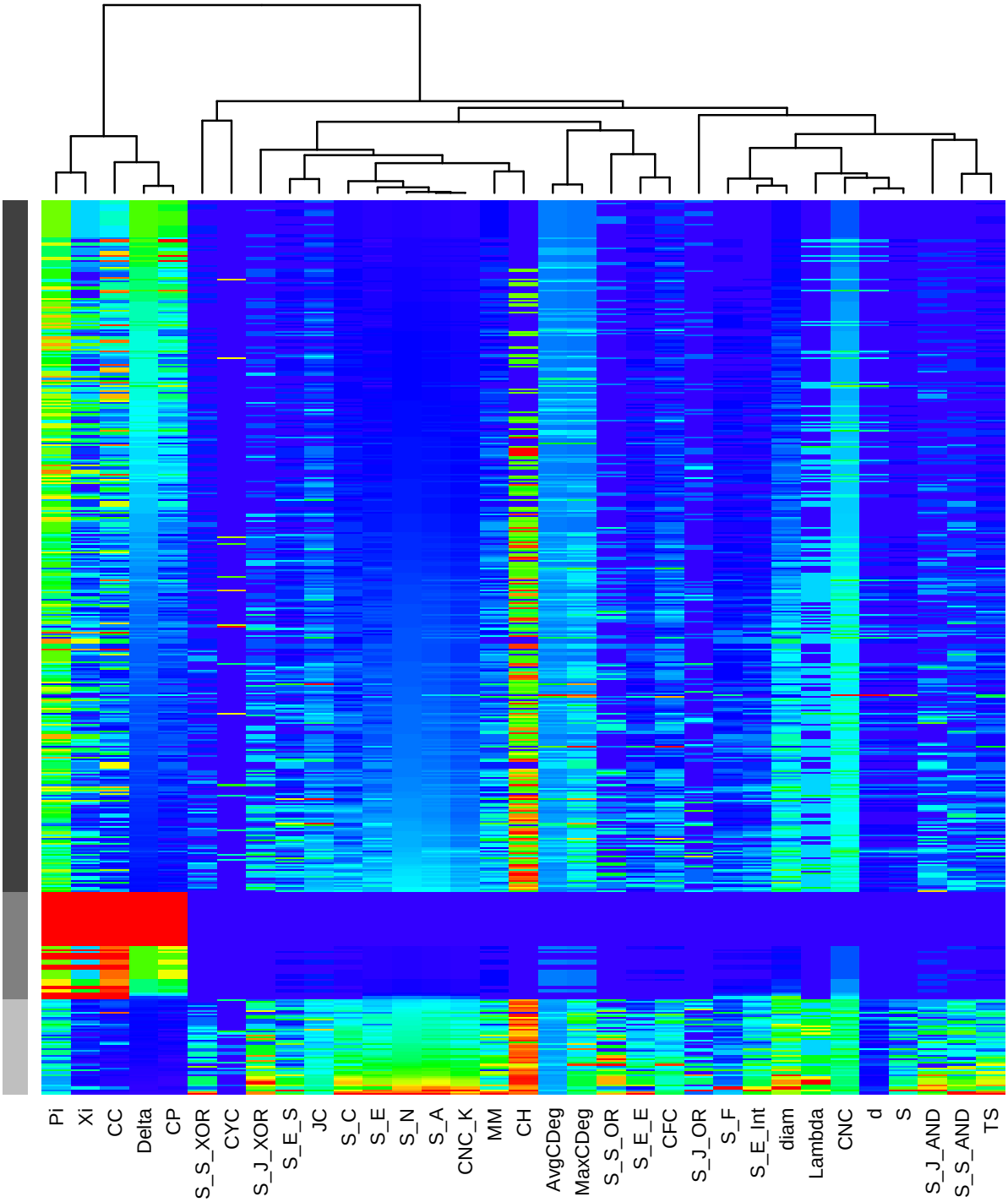

Figure 6: Clustered heatmap displaying 33 process metric values for 515 processes. The rows are separated into three clusters (see bar with gray scale values at the left). The columns are hierarchically clustered using $1 -$ the Spearman's rank correlation coefficient as distance measure.

[9] Jan Mendling. Testing density as a complexity metric for EPCs. Technical Report JM-2006-11-15, Vienna University of Economics and Business Administration, 2006.

[10] Jan Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. PhD thesis, Vienna University of Economics and Business Administration, 2007.

[11] Michael J. Panik. *Advanced Statistics from an Elementary Point of View*. Elsevier Academic Press, 2005.

[12] Andy Pryke, Sanaz Mostaghim, and Alireza Nazemi. Heatmap visualization of population based multi objective algorithms. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization: Proceedings of the 4th International Conference EMO 2007*, volume 4403 of *LNCS*, pages 361–375, 2007.

[13] Boudewijn van Dongen, Remco Dijkman, and Jan Mendling. Measuring similarity between business process models. In Zohra Bellahsène and Michel Léonard, editors, *Advanced Information Systems Engineering: Proceedings of the 20th International Conference CAiSE 2008*, volume 5074 of *LNCS*, pages 450–464, 2008.

[14] Irene Vanderfeesten, Jorge Cardoso, and Hajo A. Reijers. A weighted coupling metric for business process models. In Johann Eder, Stein L. Tomassen, Andreas L. Opdahl, and Guttorm Sindre, editors, *Proceedings of the CAiSE'07 Forum at the 19th International Conference on Advanced Information Systems Engineering*, pages 41–44, 2007.

[15] Irene Vanderfeesten, Hajo A. Reijers, Jan Mendling, Wil M. P. van der Aalst, and Jorge Cardoso. On a quest for good process models: The cross-connectivity metric. In Zohra Bellahsène and Michel Léonard, editors, *Advanced Information Systems Engineering: Proceedings of the 20th International Conference CAiSE 2008*, volume 5074 of *LNCS*, pages 480–494, 2008.