

bupaR: Enabling reproducible business process analysis

Non Peer-reviewed author version

JANSSENSWILLEN, Gert; DEPAIRE, Benoit; SWENNEN, Marijke; JANS, Mieke & VANHOOF, Koen (2018) bupaR: Enabling reproducible business process analysis. In: Knowledge-based systems, 163, p. 927-930.

DOI: 10.1016/j.knosys.2018.10.018

Handle: <http://hdl.handle.net/1942/27485>

# bupaR: Enabling Reproducible Business Process Analysis

Gert Janssenswillen<sup>a,b</sup>, Benoît Depaire<sup>a</sup>, Marijke Swennen<sup>a</sup>, Mieke Jans<sup>a</sup>,  
Koen Vanhoof<sup>a</sup>

<sup>a</sup> *Faculty of Business Economics  
UHasselt - Hasselt University  
Martelarenlaan 42 3500 Hasselt, Belgium*  
<sup>b</sup> *Flemish Research Foundation  
Egmontstraat 6 1000 Brussels, Belgium*

---

## Abstract

Over the last decades, the field of process mining has emerged as a response to a growing amount of event data being recorded in the context of business processes. Concurrently with the increasing amount of literature produced in this field, a set of tools has been developed to implement the various algorithms and provide them to end users. However, the majority of tools does not provide the possibility of creating workflows which can be reused at a later point in time to reproduce the results, and most tools are not easily customizable. This paper introduces bupaR, an integrated collection of R-packages which creates a framework for reproducible process analysis in R and supports different steps of a process analysis project, from data extraction to data analysis. It is an extensible framework of several R-packages to analyse process data, each with their specific purpose and set of tools.

*Keywords:* event data, process analysis, R, bupaR, edeaR, eventdataR, processmapR, processmonitR, xesreadR

---

## 1. Introduction

Over the last decades, the field of process mining has arisen as a response to a growing amount of event data being recorded in the context of business processes. Pioneering works considered the discovery of process models from event data [1, 2, 3], known as *process discovery*. Insights from the analyses soon proofed to be highly beneficial for companies to improve performance

7 and quality, which has caused an enormous volume of process analytics re-  
8 search, encompassing a wide range of techniques and algorithms to analyse  
9 event data [4].

10 Concurrently with the increasing amount of literature produced in this  
11 field, a set of tools has been developed to implement the various algorithms  
12 and provide them to end users. The tools that were developed are both aca-  
13 demic and commercial in nature, and are diverse concerning their customiz-  
14 ability, implementation platform or architecture, and the techniques they  
15 support. However, the existing tools have several drawbacks. Firstly, the  
16 majority of tools do not provide the possibility of creating workflows which  
17 can be reused at a later point in time to reproduce the results. Secondly,  
18 since they aim to support any possible process, most tools are not (easily)  
19 customizable, by adequately taking into account custom data attributes, or  
20 by allowing visualization to be customized according to the process context  
21 or sector. Finally, the majority of tools are stand-alone programs, solely sup-  
22 porting process mining techniques, without an interface to general-purpose  
23 data mining, visualization or statistical tools.

24 This paper introduces bupaR, a collection of R-packages which provide a  
25 framework for reproducible process analysis in R. The packages implement  
26 a class for event data in R, together with a set of generics and methods  
27 to handle it. By providing support for process analysis in R, it is the first  
28 tool to analyse processes using reusable scripts as well as to combine scripts,  
29 meta-data and interpretation of the results with Rmarkdown documents.  
30 The framework currently contains techniques for exploratory and descriptive  
31 event data analysis, for visualizing process data with process maps, and for  
32 creating real-time process monitoring dashboards, among other things.

## 33 2. Problems and Background

34 Process mining originated at the end of the 20<sup>th</sup> century with the develop-  
35 ment of algorithms trying to learn models from event data [1, 2, 3]. Over the  
36 years, many more advanced algorithms for process discovery were developed,  
37 such as the heuristics miner [5], ILP miner [6] and inductive miner [7]. Next  
38 to process discovery, conformance checking emerged as another important  
39 research track within process mining [8]. The latter is focused on the rela-  
40 tion between event data on the one hand and the process model on the other  
41 hand. It aims at finding inconsistencies between the two and furthermore

Package	Version	Functionality
bupaR [13]	0.4.0	Creation and handling of event log objects and basic preprocessing tasks
edeaR [14]	0.8.0	Calculate descriptive process metrics
eventdataR [15]	0.2.0	Contains example event data
xesreadR [16]	0.2.2	Read and write .XES-files
processmapR [17]	0.3.1	Draw process map and other process specific visualization
processanimateR [18]	0.1.1	Animate process maps
petrinetR [19]	0.1.0	Read and handle Petri Nets
processmonitR [20]	0.1.0	Create interactive dashboards for process analysis

Table 1: Current packages in the bupaR framework.

assesses the performance of discovery algorithms in their attempt to find a good representation of the process captured with the event data.

Although process discovery and conformance checking are still important topics in the process mining domain, it has recently grown much bigger. Currently, attention is given to real-time process analysis [9], blockchain [10], Internet-of-things [11], and predictive process monitoring [12], among others. The focus of bupaR currently is on the sub domain of *process analytics*, focusing entirely on the analysis of process data, and is less concerned with executable process models. In this sense, it is similar to most commercial process analysis tools.

### 3. Software Architecture and Functionalities

An overview of the different packages contained by the bupaR framework is given in Table 1. Note that the name *bupaR* refers to the overall framework as well as to the central package for supporting event data. We will generally use the term to refer to the overall framework, unless we explicitly stated otherwise. In the next paragraphs, the functionalities of each of the packages is discussed in more detail.

*bupaR*. The `bupaR`-package [13] is the core package of the framework, implements an S3-objects class for event data. It provides functions to create these objects, as well as support for common transformations. Auxiliary functions to seamlessly change the classifiers of the event data are made available, and

63 event log versions of common `dplyr` [21] functions for data manipulation are  
64 implemented, such as `filter`, `group_by` and `mutate`, among others. These  
65 functions can be used to preprocess event data. Some specific preprocessing  
66 tasks are supported explicitly by specific functions, such as aggregations of  
67 activity labels.

68 *edeaR*. `edeaR` [14] stands for Exploratory and Descriptive Event-Data Anal-  
69 yses, and contains a set of process metrics to describe and explore event logs.  
70 The process metrics are based on Lean Six Sigma literature [22] and can be  
71 analyzed and visualized at different levels of granularity. Additionally, `edeaR`  
72 contains an extensive collection of event data specific filters.

73 *eventdataR*. `eventdataR` [15] is a data-package which provide easy access to  
74 event logs for testing and experiments. Currently, both artificial event data,  
75 e.g. `patients`, as well as real-life event data, such as the Sepsis dataset [23].

76 *xesreadR*. In order to be compatible with the eXtensible Event Stream IEEE  
77 standard [24], the `xesreadR` package [16] allow to read and write .xes-files.

78 *processmapR*. Process data specific visualizations, such as process maps and  
79 dotted charts [25], are provided by `processmapR` [17]. As a result, `processmapR`  
80 is complementary to `edeaR` for exploring and describing process data, where  
81 the latter focuses more on numeric result and `processmapR` on visualizations.

82 *processanimateR*. By extending `processmapR`, `processanimateR` [18] allows  
83 to easily animate process maps using token replay.

84 *processmonitR*. In order to facilitate the creation of dashboards using Shiny  
85 [26], `processmonitR` [20] provides a limited set of process dashboards, fo-  
86 cussed on a specific aspect, e.g. performance, resources, etc. These can be  
87 used in a permanent, real-time fashion, as well as for interactive data analy-  
88 sis. While still in an experimental phase, the goal is to extend this package to  
89 allow for easy building of custom process dashboards. Furthermore, built-in  
90 support for online analysis using partial cases and using event streams can  
91 be added in the future.

92 *petrinetR*. While all the package above are centered around process data,  
93 `petrinetR` [19] is the first package to introduce a notion of process models in  
94 R. Currently, the main functionality is to create, read and write Petri Nets,  
95 to adjust them, visualize them, but also to perform token replay and parse

96 transition sequences. In future, the goal is the link this package with the  
97 other packages by means of process discovery and conformance checking.

#### 98 4. Comparison with other process analysis tools

99 In comparison with existing tools for process analytics, both open-source  
100 and commercial, bupaR can be seen is unique as it 1) is easily extensible  
101 and combinable with other tools, 2) allows to reproduce workflows, and is 3)  
102 interactive, supporting and iterative and dynamic user interaction [27].

103 One of the most extensive and open-source process mining framework to  
104 date is ProM [24]. It contains most of the state-of-the-art techniques which  
105 are developed in related literature. It can be extended with java-libraries,  
106 although it requires a considerable time investment to do so, as one has to  
107 be familiar with the source-code of the central frame-work. Furthermore,  
108 its setup, with a click-and-select user interface, makes it hard to make your  
109 analysis reproducible. In order to enable reproducible workflows, Rapid-  
110 ProM [28], an extension to RapidMiner, was developed. RapidProM allows  
111 the execution of the most widely used ProM-plugins within RapidMiner. As  
112 a result, RapidProM supports reproducible process analysis workflows, using  
113 the RapidMiner GUI of dropping and connect operators, and provides an in-  
114 terface with all the other data analysis techniques available in RapidMiner.  
115 Also the interactiveness of RapidProm is rather low, as assembling a work-  
116 flow typically requires a clear goal decided upon beforehand, and altering  
117 workflows can be cumbersome. Other commercial tooling score higher on  
118 interactiveness, especially due to the use of interactive graphical visualiza-  
119 tions (e.g. Disco<sup>1</sup>, Celonis<sup>2</sup>). However, reproducibility and extensibility is  
120 generally very low.

121 Some support for event data in its broadest sense is already available in  
122 the form of several R packages. For instance, the events package [29] uses the  
123 KEDS (Kansas Event Data System) format [30]. This format is targeted at  
124 political event data, and typically extracted from news reports. In addition,  
125 *eventstudies* [31] regards event data as a dataset with two columns, *name* and  
126 *when*. It thus contains information about when a specific event happened for  
127 a certain subject. It is clear that none of these existing packages support the

---

<sup>1</sup><https://fluxicon.com/disco/>

<sup>2</sup><https://www.celonis.com/>

128 more complex data structures typical for business process data, nor do they  
129 provide the required tools to analyze these.

## 130 5. Applications and Illustrative Examples

131 The bupaR framework has been applied in academic works such as [32],  
132 project such as the H2020 project HUMAN<sup>3</sup>, as well as by professionals<sup>4</sup>

133 Some examples of functionalities are shown in Figure 1, which can be cre-  
134 ated using the R statements below. Figure 1a shows a process map, colored  
135 according to the processing time of activities. Figure 1b shows a dotted chart,  
136 which displays how activities are distributed along the time of day. Figure  
137 1c shows a resource-activity matrix, where one can observe which resources  
138 executed which activities. The data used in these examples can be found in  
139 the eventdataR package. For more examples, we refer to the documentation  
140 and website.<sup>5</sup>

```
141 #Example a  
142 process_map(patients, type = performance())  
143 #Example b  
144 dotted_chart(sepsis, x = "relative_day", y = "start_day")  
145 #Example c  
146 resource_frequency(sepsis,  
147   level = "resource-activity") %>% plot
```

## 148 6. Conclusions

149 In this paper, we introduced a collection of R-packages which were de-  
150 signed to support the different analytical stages within process analysis, from  
151 the data extraction to the analysis and mining. It is the first effort to support  
152 the handling and analysis of process event data in R.

153 Making process analysis possible in R will improve the reproducibility  
154 of process analyses. Reusable analysis scripts can be combined with the  
155 interpretation of the analysis as well as with meta-data. Furthermore, it will

---

<sup>3</sup><http://humanmanufacturing.eu>

<sup>4</sup><https://medium.com/@gscheithauer/process-mining-in-10-minutes-with-r-1ab28ed74e81>

<sup>5</sup><http://bupar.net>

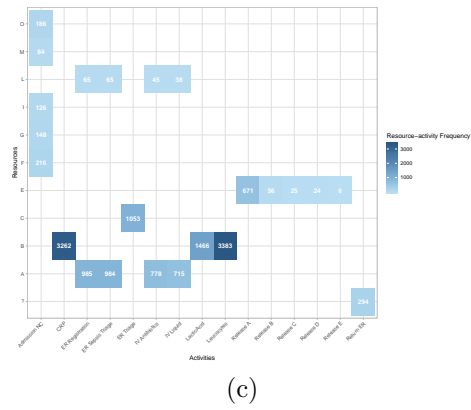
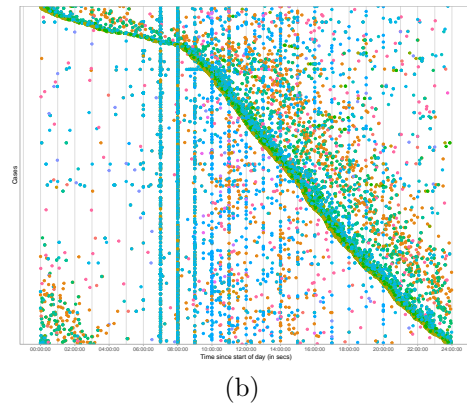
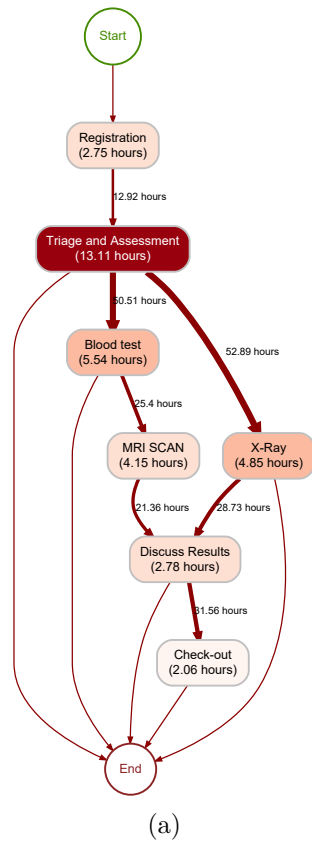


Figure 1: Examples of process visualizations



allow process analysts to easily create custom analysis tools, and will enlarge the adoption and publicity of process mining in industry.

Further extensions to the framework are planned for the near future, in order to resolve some important limitations of current functionalities. Foremost, the support for working with executable process models in R, such as Petri Nets and BPMN models should be improved. Subsequently, we believe that providing process discovery algorithms and conformance checking are important next steps, in order to support end-to-end process analysis. The best way to do this, by reimplementing existing approaches, or by creating interfaces with other tools, still has to be decided upon.

## References

- [1] J. E. Cook, A. L. Wolf, Automating process discovery through event-data analysis, in: Software Engineering, 1995. ICSE 1995. 17th International Conference on, IEEE, 1995, pp. 73–73.
- [2] R. Agrawal, D. Gunopulos, F. Leymann, Mining process models from workflow logs, in: H. J. Schek, F. Saltor, I. Ramos, G. Alonso (Eds.), Advances in Database Technology - EDBT '98, Vol. 1377, Springer-Verlag Berlin Heidelberg, 1998, pp. 467–483.
- [3] A. Datta, Automating the discovery of as-is business process models: Probabilistic and algorithmic approaches, Information Systems Research 9 (3) (1998) 275–301.
- [4] W. van der Aalst, Process mining: discovery, conformance and enhancement of business processes, Springer, Heidelberg, 2011.
- [5] A. Weijters, J. Ribeiro, Flexible Heuristics Miner (FHM), in: 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 2011, pp. 310–317.
- [6] J. M. E. Van der Werf, B. F. van Dongen, C. A. Hurkens, A. Serebrenik, Process discovery using integer linear programming, in: Applications and Theory of Petri Nets, Springer, 2008, pp. 368–387.
- [7] S. J. J. Leemans, D. Fahland, W. M. P. van der Aalst, Discovering block-structured process models from event logs-a constructive approach, in: Application and Theory of Petri Nets and Concurrency, Springer, 2013, pp. 311–329.

- 189 [8] A. Rozinat, W. M. P. van der Aalst, Conformance checking of processes  
190 based on monitoring real behavior, *Information Systems* 33 (1) (2008)  
191 64–95.
- 192 [9] F. M. Maggi, A. Burattin, M. Cimitile, A. Sperduti, Online Process  
193 Discovery to Detect Concept Drifts in LTL-Based Declarative Process  
194 Models, in: *On the Move to Meaningful Internet Systems: OTM 2013*  
195 *Conferences, Lecture Notes in Computer Science*, Springer, Berlin, Hei-  
196 delberg, 2013, pp. 94–111.
- 197 [10] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev,  
198 J. Mendling, Untrusted business process monitoring and execution using  
199 blockchain, in: *International Conference on Business Process Manage-*  
200 *ment*, Springer, 2016, pp. 329–347.
- 201 [11] C. Janiesch, A. Koschmider, M. Mecella, B. Weber, A. Burattin,  
202 C. Di Ciccio, A. Gal, U. Kannengiesser, F. Mannhardt, J. Mendling,  
203 A. Oberweis, M. Reichert, S. Rinderle-Ma, W. Song, J. Su, V. Tor-  
204 res, M. Weidlich, M. Weske, L. Zhang, The Internet-of-Things Meets  
205 Business Process Management: Mutual Benefits and Challenges,  
206 arXiv:1709.03628 [cs]ArXiv: 1709.03628.  
207 URL <http://arxiv.org/abs/1709.03628>
- 208 [12] F. Folino, M. Guarascio, L. Pontieri, Mining predictive process models  
209 out of low-level multidimensional logs, in: *International conference on*  
210 *advanced information systems engineering*, Springer, 2014, pp. 533–547.
- 211 [13] G. Janssenswillen, bupaR: Business Process Analytics in R, r package  
212 version 0.3.0 (2017).
- 213 [14] G. Janssenswillen, M. Swennen, edeaR: Exploratory and Descriptive  
214 Event-Based Data Analysis, r package version 0.7.1 (2017).
- 215 [15] G. Janssenswillen, eventdataR: Event Data Repository, r package ver-  
216 sion 0.1.1 (2017).
- 217 [16] G. Janssenswillen, B. Depaire, xesreadR: Read and Write XES Files, r  
218 package version 0.2.1 (2017).
- 219 [17] G. Janssenswillen, processmapR: Construct Process Maps Using Event  
220 Data, r package version 0.2.0 (2017).

- 221 [18] F. Mannhardt, processanimateR: Process Map Animation (2018).
- 222 [19] G. Janssenswillen, petrinetR: Building, Visualizing, Exporting and Re-  
223 playing Petri Nets, r package version 0.1.0 (2016).
- 224 [20] G. Janssenswillen, processmonitR: Building Process Monitoring Dash-  
225 boards, r package version 0.1.0 (2017).  
226 URL <https://CRAN.R-project.org/package=processmonitR>
- 227 [21] H. Wickham, R. Franois, L. Henry, K. Mller, dplyr: A Grammar of Data  
228 Manipulation, R package version 0.7.5 (2018).  
229 URL <https://CRAN.R-project.org/package=dplyr>
- 230 [22] M. Swennen, G. Janssenswillen, M. J. Jans, B. Depaire, K. Vanhoof,  
231 Capturing Process Behavior with Log-Based Process Metrics, in: Pro-  
232 ceedings of the 5th International Symposium on Data-driven Process  
233 Discovery and Analysis (SIMPDA), Vienna, 2015.
- 234 [23] F. Mannhardt, Sepsis Cases - Event Log, dOI: 10.4121/uuid:915d2bfb-  
235 7e84-49ad-a286-dc35f063a460 (2016).
- 236 [24] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. Van Dongen, W. M. P. Van  
237 Der Aalst, Xes, xesame, and prom 6, in: P. Soffer, E. Proper (Eds.),  
238 Information Systems Evolution, Springer, 2011, pp. 60–75.
- 239 [25] M. Song, W. M. van der Aalst, Supporting process mining by showing  
240 events at a glance, in: Proceedings of the 17th Annual Workshop on  
241 Information Technologies and Systems (WITS), 2007, pp. 139–145.
- 242 [26] W. Chang, J. Cheng, J. Allaire, Y. Xie, J. McPherson, shiny: Web  
243 Application Framework for R, r package version 1.1.0 (2018).  
244 URL <https://CRAN.R-project.org/package=shiny>
- 245 [27] L. Anselin, Interactive techniques and exploratory spatial data analysis,  
246 Geographical Information Systems: principles, techniques, management  
247 and applications 1 (1999) 251–264.
- 248 [28] W. M. P. van der Aalst, A. Bolt, S. J. van Zelst, RapidProM: Mine  
249 Your Processes and Not Just Your Data, arXiv:1703.03740 [cs]ArXiv:  
250 1703.03740.  
251 URL <http://arxiv.org/abs/1703.03740>

- 252 [29] W. Lowe, events: Store and manipulate event data, r package version  
253 0.5 (2012).
- 254 [30] P. A. Schrodtt, S. G. Davis, J. L. Weddle, Political science: KEDSa  
255 program for the machine coding of event data, Social Science Computer  
256 Review 12 (4) (1994) 561–587.
- 257 [31] C. Anand, V. Balasubramaniam, V. Bahure, A. Shah, eventstudies: An  
258 R package for conducting event studies and a platform for methodolog-  
259 ical research on event studies. (2014).
- 260 [32] D. Etinger, T. Orehovački, S. Babić, Applying process mining techniques  
261 to learning management systems for educational process model discovery  
262 and analysis, in: International Conference on Intelligent Human Systems  
263 Integration, Springer, 2018, pp. 420–425.

264 **Required Metadata**

265 *Current code version*

Nr.	Code metadata description	Please fill in this column
C1	Current code version	0.3.2
C2	Permanent link to code/repository used of this code version	<i>https : //github.com/cran/bupaR</i>
C3	Legal Code License	MIT-License
C4	Code versioning system used	git
C5	Software code languages, tools, and services used	R
C6	Compilation requirements, operating environments & dependencies	
C7	If available Link to developer documentation/manual	<i>http : //www.bupar.net</i>
C8	Support email for questions	<i>gert.janssenswillen@uhasselt.be</i>

Table 2: Code metadata