

Process Analysis with bupaR 0.5.0: What's New? (Extended Abstract)

Gerhardus A. W. M. van Hulzen^{1,*}, Gert Janssenswillen¹, Niels Martin^{1,2} and Benoît Depaire¹

¹Research group Business Informatics, Hasselt University, Martelarenlaan 42, 3500 Hasselt, Belgium

²Research Foundation Flanders (FWO), Egmontstraat 5, 1000 Brussels, Belgium

Abstract

bupaR and the bupaverse are a collection of open-source R-packages designed for process data analysis in R. Due to its focus on interactivity, reproducibility, and extensibility, combined with its open-source nature, bupaR has seen a significant increase in usage over the past few years, both by academics and professional process analysts. In this demonstration, we highlight the new features of bupaR 0.5.0, which can assist practitioners when analysing their process data.

Keywords

bupaR, R, Process analytics, Process mining, Event data

1. Introduction

Several open-source software solutions are available for process mining analyses, such as ProM [1], PM4Py [2], Apromore CE [3], and bupaR [4]. The availability of these tools allows professionals to experiment and experience the value of process mining easily and free of charge.

For process and data analysts familiar with the statistical software environment R [5], the bupaverse collection of R-packages provide a starting point for the analysis of process data. The core focus of bupaverse is based on three key principles: (i) extensibility, (ii) reproducibility, and (iii) interactivity [4, 6]. These fundamental principles, together with its open-source nature, have contributed to its widespread use.

We continuously improve and add new features to enhance the functionalities offered by bupaverse. This paper presents the release highlights of bupaR 0.5.0 [7], discusses its maturity and how one can start using it, and briefly looks forward to future development and releases.

ICPM 2022 Doctoral Consortium and Tool Demonstration Track


*Corresponding author.

✉ gerard.vanhulzen@uhasselt.be (G. A. W. M. van Hulzen); gert.janssenswillen@uhasselt.be (G. Janssenswillen); niels.martin@uhasselt.be (N. Martin); benoit.depaire@uhasselt.be (B. Depaire)

🆔 0000-0001-8962-9515 (G. A. W. M. van Hulzen); 0000-0002-7474-2088 (G. Janssenswillen); 0000-0003-3279-3853 (N. Martin); 0000-0003-4735-0609 (B. Depaire)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

2. New Features

2.1. Activity Log

In bupaR 0.5.0, a new kind of log format has been introduced: the *activity log*. In an activity log, each row represents a single activity instance. This means that, as opposed to an event log in which each row represents an event occurring at a particular point in time, an activity log can have multiple timestamps per row (e.g. schedule, start, complete, etc.) [8, 9]. These are stored across multiple columns, in contrast to the single timestamp column of an event log. An example of conversion between event log to activity log and vice versa is shown in Fig. 1.

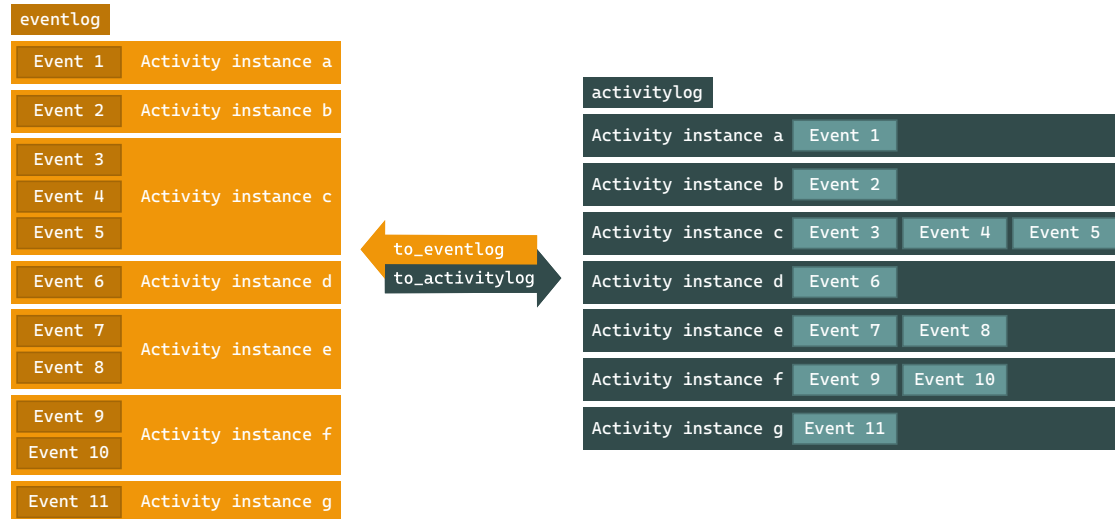


Figure 1: Conversion from eventlog to activitylog, and vice versa [7].

The activity log has been implemented as a new `S3` class object (`activitylog`) alongside the existing `eventlog` object. The main advantages of the new `activitylog` object are a reduced memory footprint and increased analysis performance. Especially for analyses on activity instance level, e.g. the durations of activities, the new `activitylog` is more convenient and efficient because all events belonging to the same activity instance are stored on the same entry in the log. Moreover, activity attributes are recorded only once per activity instance, instead of repeatedly for each event of the same instance.

Nevertheless, this does not imply that `eventlog` is completely superseded. In fact, the `eventlog` provides more flexibility because attributes can be stored at the event level, allowing events of the same activity instance to have different attributes. For example, different resources could be responsible for the start and completion of an activity instance. In addition, in an `eventlog`, the same lifecycle (e.g. schedule, start, complete, etc.) can be repeated multiple times, which is useful when the activity instance was suspended and later resumed. Therefore, depending on the use case, either `eventlog` or `activitylog` is the most appropriate format. Currently, `bupaR`, `edear`, `processmapR`, and `processcheckR` fully support `activitylog` objects, and

other bupaverse packages will follow in subsequent releases. Moreover, logs can be conveniently transformed from one into the other using the `to_eventlog()` and `to_activitylog()` functions.

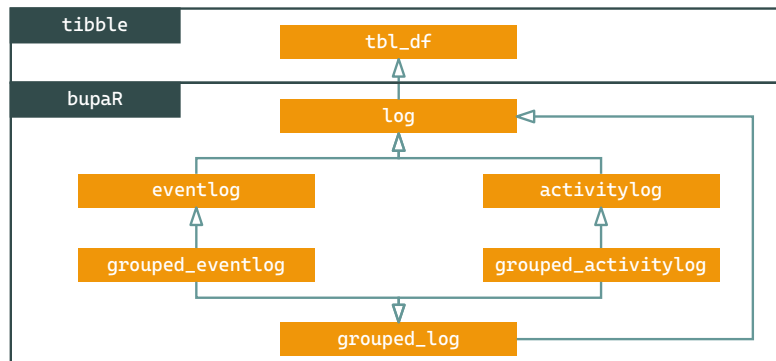


Figure 2: bupaR S3 class inheritance schema.

In order to implement `activitylog` and facilitate the extensibility of the bupaR ecosystem, we have revised the S3 class inheritance of `log` objects. Fig. 2 visualises the new class inheritance schema. Both `eventlog` and `activitylog` are inherited from the new base `log` class, which in turn uses a `tbl_df` from the `tibble` package [10] as back-end data storage. When grouping is applied to a `log` class using the `group_by()` functions, it becomes a `grouped_log` to signify the presence of grouping variable(s).

2.2. Augmenting Logs

As of `edeR` 0.9.0, our package for exploratory and descriptive event data analysis, all `append` and `append_column` arguments of descriptive metrics (e.g. `activity_frequency()`, `processing_time()`, etc.) have been deprecated in favour of a new `augment()` method, which is consistent with the `broom` package [10] for adding outputs of predictions and estimations to data. The new workflow is visualised in Fig. 3, and a code example is provided in Listing 1. For instance, we can calculate the throughput times for each case on the `sepsis` log and add these times back to the `sepsis` log as a new column `"case_throughput_time"`.



Figure 3: Augmenting a log [7].

```

1 sepsis %>%
2   throughput_time(level = "case") %>%
3   augment(log = sepsis, columns = "throughput_time", prefix = "case")
  
```

Listing 1: R example of augmenting a log.

This new workflow ensures consistent separation between the outputs of descriptive metrics and `log` objects. Furthermore, the `augment()` method provides a standardised, flexible, and transparent way to enrich logs with descriptive metrics.

2.3. Improved Data Manipulation

Significant changes have been made to the supported `dplyr` [10] methods for data manipulation in `bupaR` (e.g. `filter`, `mutate`, `slice`, etc.), most significantly to `group_by()`, for grouping event data for descriptive analyses. For example, the number of cases in which each activity was executed can be calculated using the code shown on line 1 in Listing 2.

```
1 sepsis %>% group_by(activity) %>% n_cases()
2 sepsis %>% group_by_ids(activity_id) %>% n_cases()
3 sepsis %>% group_by_activity() %>% n_cases()
```

Listing 2: R example of `group_by`.

A more convenient way of grouping `log` objects as of `bupaR` 0.5.0 is by using the `group_by_ids()` method, completed with the desired `bupaR` attribute function(s) (e.g. `activity_id`, `case_id`, etc.), or by directly using `group_by_activity()`, as shown on lines 2 and 3, respectively. These new grouping methods allow conducting grouped descriptive analyses more conveniently without knowing the underlying column names. Moreover, the handling of grouped logs is improved so that any metric can now be computed for any (set of) grouping variable(s).

3. Maturity & Usage

Since its conception, `bupaR` has received over 800K downloads in over 160 countries. Users come from various industries, e.g., healthcare, governance, automotive, and academics. Stable versions of `bupaR` and other `bupaverse` packages can be installed from CRAN using `install.packages("bupaverse")` or, for the version with the latest patches and bugfixes, directly from GitHub¹ using `devtools::install_github("bupaverse/bupaverse")`. A demonstration of the release can be found here.² Furthermore, the `bupar.net` website contains ample documentation and examples on `bupaR` and the `bupaverse` packages.

4. Conclusion & Future Work

This paper presented the release highlights of `bupaR` 0.5.0, most notably the introduction of the activity log, a new standardised way to augment logs, and improved data manipulation.

Future releases will focus on extending the `bupaverse` ecosystem with new functionalities for process analysis and maintenance of existing code. New functionalities, such as Performance Spectrum [11], trace and activity clustering, social network mining and process discovery, are currently on the roadmap. Other functionalities can be requested using GitHub Issues.¹

¹<https://github.com/bupaverse/>

²<https://tinyurl.com/icpmdemobupar>

Acknowledgments

The authors would like to warmly thank all users who are actively contributing to the bupaR-framework by submitting issues and pull requests on the GitHub¹ repositories.

This study was supported by the Special Research Fund (BOF) of Hasselt University under Grant No. BOF19OWB20.

References

- [1] B. F. van Dongen, A. K. A. de Medeiros, E. H. M. W. Verbeek, A. J. M. M. Weijters, W. M. P. van der Aalst, The ProM Framework: A New Era in Process Mining Tool Support, volume 3536 of *LNCS*, Springer, 2005, pp. 444–454. doi:10.1007/11494744_25.
- [2] A. Berti, S. J. van Zelst, W. M. P. van der Aalst, Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science, volume 2374 of *CEUR Workshop Proceedings*, 2019, pp. 13–16.
- [3] M. La Rosa, H. A. Reijers, W. M. P. van der Aalst, R. M. Dijkman, J. Mendling, M. Dumas, L. García-Bañuelos, APROMORE: An Advanced Process Model Repository, *Expert Syst. Appl.* 38 (2011) 7029–7040. doi:10.1016/j.eswa.2010.12.012.
- [4] G. Janssenswillen, B. Depaire, M. Swennen, M. J. Jans, K. Vanhoof, bupaR: Enabling Reproducible Business Process Analysis, *Knowl. Based Syst.* 163 (2019) 927–930. doi:10.1016/j.knosys.2018.10.018.
- [5] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, 2022. URL: <https://www.R-project.org>.
- [6] G. Janssenswillen, F. Mannhardt, M. Creemers, B. Depaire, L. Jooken, N. Martin, G. Van Houdt, Extensions to the bupaR Ecosystem: An Overview, volume 2703 of *CEUR Workshop Proceedings*, 2020, pp. 43–46.
- [7] G. Janssenswillen, bupaR 0.5.0: What’s new?, 2022. URL: <https://bupaR.net/2022/07/27/bupaR-0-5-0-whats-new/>.
- [8] N. Martin, G. Van Houdt, G. Janssenswillen, DaQAPO: Supporting Flexible and Fine-Grained Event Log Quality Assessment, *Expert Syst. Appl.* 191 (2022) 116274. doi:10.1016/j.eswa.2021.116274.
- [9] L. Bouarfa, J. Dankelman, Workflow Mining and Outlier Detection from Clinical Activity Logs, *J. Biomed. Inform.* 45 (2012) 1185–1190. doi:10.1016/j.jbi.2012.08.003.
- [10] H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Golemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, H. Yutani, Welcome to the Tidyverse, *J. Open Source Softw.* 4 (2019) 1686. doi:10.21105/joss.01686.
- [11] V. Denisov, E. Belkina, D. Fahland, W. M. P. van der Aalst, The Performance Spectrum Miner: Visual Analytics for Fine-Grained Performance Analysis of Processes, volume 2196 of *CEUR Workshop Proceedings*, 2018, pp. 96–100.