Extensions to the bupaR ecosystem: An overview

Gert Janssenswillen*, Felix Mannhardt†, Mathijs Creemers*, Benoît Depaire*,
Mieke Jans*, Leen Jooken*, Niels Martin*‡ and Greg Van Houdt*

*UHasselt - Hasselt University
Agoralaan, 3590 Diepenbeek, Belgium
gert.janssenswillen@uhasselt.be

†Technische Universiteit Eindhoven
5612 AZ Eindhoven, Netherlands
f.mannhardt@tue.nl

‡Research Foundation Flanders (FWO)
Egmonstraat 5, 1000 Brussel, Belgium

***Abstract*—Over the past few year, `bupaR` — the open-source R-ecosystem for process analysis — has seen a considerable increase in functionalities and users. It has been one of the first successful tools for script-based process analytics, and can currently be seen as the state-of-the-art tool for process analysis in R and an important player in the open-source process mining tool landscape. With a user-base consisting largely of professional process analysts, the ecosystem has helped to increase the adoption of process mining in a broad range of fields. In this demonstration, we highlight recent extensions to the ecosystem that will further increase its usefulness for practitioners during their process mining projects.**

***Index Terms*—`bupaR`, `R`, process analytics, data quality, knowledge management.**

I. Introduction

`bupaR` is an ecosystem of R-packages geared towards the analysis of process data in R [1].The ecosystem builds upon three key principles: (1) connectivity, (2) reproducibility and (3) extensibility. The latter indicates that the functionalities provided by `bupaR` are continuously evolving. Since the release of the core packages in 2017, both its usage and the range of provided functionalities have been steadily increasing. As shown in Table I, `bupaR` currently consists of 16 interconnected libraries for process analysis in the ecosystem, each targeting a specific problem or use case.

While `bupaR` in itself is not new, this paper outlines a significant number of new functionalities that have recently been added to the ecosystem. Hence, the current paper extends earlier publications about the functionalities for business process analysis in R [1], [2].

This paper is organised as follows. Section II lists recently developed functionalities, Section III discusses the maturity and usage of `bupaR`, while Section IV concludes the paper. An accompanying tutorial and screencast can be found on GitHub.[1]

[1]https://github.com/bupaverse/icpm-demo-tutorial

TABLE I
Overview of bupaR-ecosystem.

Packages	Purpose
bupaR*	Core event log functionalities
collaborateR	Create Collaboration Graphs
daqapo*	Identify data quality issues in process-oriented data
edeaR*	Exploratory and descriptive event data analysis
eventdataR*	Repository of event logs
heuristicsmineR*	Discover models using the Heuristics Miner
logbuildR	Facilitate event log construction
pm4py*	Bridge with the PM4Py python library
processanimateR*	Animate process maps
processcheckR*	Rule-based conformance checking
processmapR*	Create process maps
processmonitR*	Create process monitoring dashboards
propro	Create probabilistic process models
petrinetR*	Support for petri nets
understandBPMN*	Calculate understandability metrics for BPMN
xesreadR*	Read and write XES-files

*Published on CRAN (https://cran.r-project.org/)

II. New features

A. LogbuildR

Getting event data in the right format before starting your analyses remains one of the important hurdles that process analysts have to take. Notwithstanding `bupaR`'s functionality for reading event logs from XES-files [3], practitioners typically have to start from raw data, and make sure that it is correctly converted into an event log.

In order to guide this conversion, the package `logbuildR` has been developed. It provides a graphical interface that leads the user through different steps to build an event log. The package provides the user with intelligent suggestions and direct feedback in each step, which help the analyst to select appropriate identifiers (case, activity, etc), make sure that each row represents a unique event in the process (versus multiple timestamps per row), convert timestamps to appropriate data formats, and ensure life-cycle values adhere to the agreed-upon

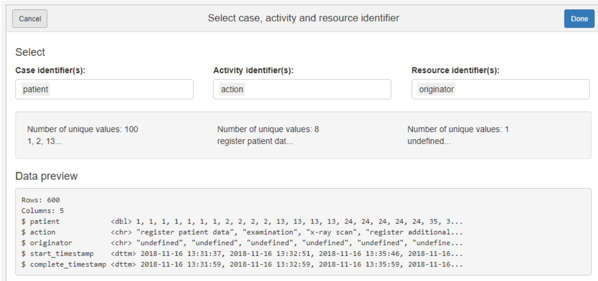

Fig. 1. Example of `logbuildR` interface: selecting appropriate identifiers.

standard transactional life-cycle model [4]. A screenshot of the graphical interface is shown in Figure 1. The `logbuildR` package is available on GitHub.[2]

B. DaQAPO

Following the preparation of the event log, one of the first steps in process analysis is to assess the quality of the data. In order to support this step, `daqapo` was developed [5], [6]. Short for Data Quality Assessment for Process-oriented Data, `daqapo` provides a variety of methods to detect data quality issues in process-oriented data.

As the reliability of process analysis techniques largely depends on the quality of the event log, data quality is an important aspect to consider. Insufficient data quality, or an inadequate understanding of it, will inevitably lead to low-quality results — *Garbage in, garbage out* — or even misleading ones — *Garbage in, gospel out*.

In order to stress the importance of data quality, `daqapo` provides a large set of checks which enable users to identify a range of data quality issues in a systematic way. These issues include missing events, incorrect timestamps, and inaccurate resource information. An overview of the available functions is shown in Table II. The `daqapo` package is available for installation on the Comprehensive R Archive Network (CRAN).[3]

C. HeuristicmineR

The package `heuristicsmineR` brings extensible support for variants of the Flexible Heuristics Miner [7] to bupaR. Two major variants are implemented: the original Flexible Heuristics Miner as described in [7] and a variant that uses time intervals derived from life-cycle transitions as described in [8]. Having discovered a Causal net, the dependencies and gateway information can be visualised or transformed into a Petri net for further processing, e.g., by computing alignments with the `pm4py` package — which bridges the `bupaR`-ecosystem to the `PM4Py` python library for process mining. An underlying design principle of the package is to separate the computation into several phases, each of which provides an intermediate result that can be inspected and visualised using the standard R `print` functionality. This makes this package well-suited for a teaching context in which the computations are followed in a step-wise fashion. Also, it is easy to compose new variants based on different heuristics.

D. Propro

The results of control-flow discovery algorithms are mainly deterministic process models, which do not convey a notion of probability or uncertainty. Using Bayesian inference and Markov Chain Monte Carlo, `propro` [9] can build a statistical model on top of a process model using event data, which is able to generate probability distributions for choices in a process' control-flow. `propro` is based on a generic algorithm to build a statistical model [10], which can then be used to test different kinds of hypotheses, such as non-deterministic dependencies between different choices in the model. This leads to valuable information about the process under consideration, which go beyond the discovery of its static control-flow. Hence, `propro` supports the enhancement of discovered process models by exposing probabilistic dependencies, and allows to compare the goodness-of-fit of different models with respect to the event data, each of which provides important advancements in the field of process mining. The `propro` package is available on GitHub.[4]

E. ProcessanimateR

Animation using moving tokens can be a powerful visualisation tool to help understand the general process behavior. The package `procesanimateR` implements an animation library for `bupaR` that renders interactive process animations using the web standard SVG.

In `procesanimateR`, each case is represented by a separate token that moves along the process map with speed relative to the observed activity processing and waiting times. The visual appearance of tokens can be customised using any SVG shape and core properties, such as size and color, and can be dynamically adjusted based on event attributes. In a recent release, the package was extended with support to project discovered process maps to an interactive geographical map in which each process activity has a fixed position, as shown in Figure 2. This enables new forms of animation and process visualisation in which the position of activities and the length of edges are assigned clear semantics. This contrasts to the often random placement of activities and edges in traditional process visualisation tools.

F. CollaborateR

Whereas most functionalities of `bupaR` have been developed with no specific type of process in mind, this can not be said about `collaborateR` [11]. The origin of this package lies in the area of software engineering. As its name implies, it focuses on the collaboration between different process participants. The underlying algorithm was published in recent previous work [12].

In the fast-changing and flexible software engineering environments of today, knowledge management is critical. A clear

[2]https://github.com/bupaverse/logbuildR
[3]https://cran.r-project.org/package=daqapo
[4]https://github.com/bupaverse/propro

TABLE II
AVAILABLE ASSESSMENT FUNCTIONS IN DAQAPO.

Function	Description
`detect_activity_frequency_violations`	Detect case-wise anomalies in the number of occurrences of activities.
`detect_activity_order_violations`	Detect violations in the order of activities within cases.
`detect_attribute_dependencies`	Detect event-wise violations between attributes using logical conditions.
`detect_case_id_sequence_gaps`	Detect gaps in case identifiers, i.e. when case identifier is a numerical id.
`detect_conditional_activity_presence`	Detect activity presence versus logical conditions
`detect_duration_outliers`	Detect activity duration outliers
`detect_inactive_periods`	Detect inactive periods, i.e. periods without new arriving cases, or periods without any activity instances.
`detect_incomplete_cases`	Detect incomplete cases, given a set of *essential* activities, or *final* activities in the process.
`detect_incorrect_activity_names`	Detect incorrect activity names
`detect_missing_values`	Detect missing values
`detect_multiregistration`	Detect multi-registration, i.e. events recorded at the same time which belong to the same case or the same resource.
`detect_overlaps`	Detect overlapping activity instances
`detect_related_activities`	Detect missing related activities, i.e. when certain activities should co-exist.
`detect_similar_labels`	Detect spelling mistakes by searching for similar labels in a column.
`detect_time_anomalies`	Detect time anomalies, i.e. activities with a negative and/or zero duration.
`detect_unique_values`	Search for unique combinations of a given set of columns.
`detect_value_range_violations`	Detect invalid values, for categorical, numeric as well as time attributes.

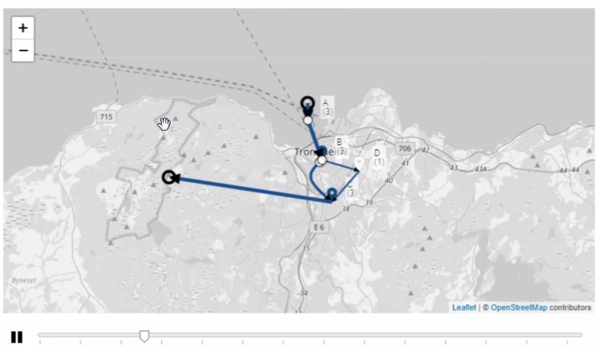

Fig. 2. Screenshot of a process animation where the process map has been projected on a geographical map.

overview on how software developers collaborate can unearth valuable patterns such as the general structure of collaboration, crucial resources, and risks (e.g. losing certain knowledge when a programmer decides to leave the company). Version control system (VCS) logs, which keep track of which tasks team members work on and when, contain data to provide these insights. `collaborateR` provides an algorithm which extracts and visualises a collaboration graph from VCS log data. The algorithm is partly based on the principles that also underlie the Fuzzy Miner [13]. Its structure consists of four phases: (1) building the base graph, (2) calculating weights for nodes and edges, and (3) simplifying the graph using aggregation and abstraction. Each of these phases offers the user flexibility to decide which parameters and metrics to include. This makes it possible for the human expert to exploit her existing knowledge about the project and team to guide the algorithm in building the graph that best fits the specific use case, and hence will provide the most accurate insights.

An example of a collaboration graph is shown in Figure 3. In this graph, pink nodes are individual programmers, while blue nodes are clusters of programmers. When programmers have worked on the same files of the project, i.e. the same software code, an edge is drawn between them. The colouring of the edges indicates whether programmers worked separately (orange), together using pair programming (green), or a mix of both (blue). The size of both nodes and edges indicates the importance of the programmers and the strength of their relationships. The package is available on GitHub.[5]

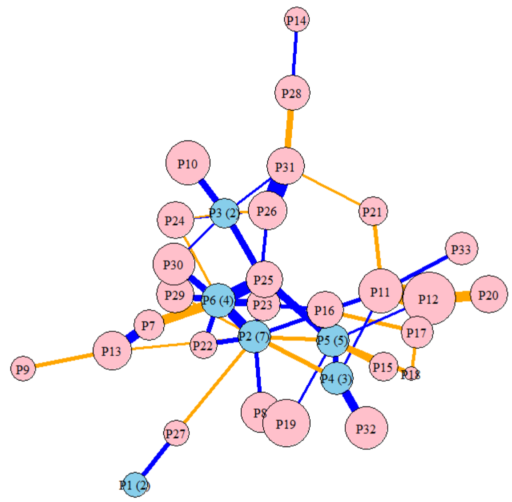

Fig. 3. Example of collaboration graph.

III. MATURITY AND USAGE

The packages of the `bupaR` collection that have been published on the Comprehensive R Archive Network (13 at the moment of writing, cf. Table I) gathered over 300k downloads - more than half of which during the past year. The tools have

[5]https://github.com/bupaverse/collaborateR

been downloaded in 140 different countries. The core packages `bupaR`, `edeaR` and `processmapR` respectively receive on average about 7k, 5k and 4k downloads each month, and are amongst the 10% most downloaded R packages.

`bupaR` has been used in general process mining research [14]–[17], and has been applied in more specific areas such as process simulation [18], transportation [19], healthcare [20], Learning Analytics [21]–[24], predictive process monitoring [25], [26], and others [27], [28]. As the majority of users are practitioners, `bupaR` has a profound impact on the adoption of process mining in various fields such as healthcare, consulting, manufacturing, telecommunications, and governmental agencies. In more popular media, various case studies are available, for example, in the context of traditional business processes, such as purchase-to-pay processes[6], how to use it with Power BI[7], or how to use it for web analytics.[8] The new functionalities described in this paper further enhance the usefulness of `bupaR` for both researchers and practitioners.

IV. Conclusion and Future Work

Since the introduction of `bupaR`, the ecosystem has steadily grown into broad toolbase, and has become widely used for process analytics. The extensions described in this paper will further enhance the use of `bupaR`, and its role in the adoption of process mining by practitioners in various industries.

Future work will focus on the extension of the new functionalities described in this paper, as well as adding new components to the eco-system. While logbuildR is now a graphical interface, it will be extended in the future so that the user will also receive the R-code that is needed to produce the event data at the end. This code can be used for scripts or reports, thereby making the log building step also reproducible. Furthermore, the creation of collaboration graphs will be generalised so that it can be used for other process data as well, beyond version control systems. New functionalities in the area of process discovery, process data visualisation and predictive process monitoring are currently being developed.

Acknowledgements

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 GitHub repositories.

References

[1] G. Janssenswillen, B. Depaire, M. Swennen, M. Jans, and K. Vanhoof, "bupar: Enabling reproducible business process analysis," *Knowledge-Based Systems*, vol. 163, pp. 927–930, 2019.

[2] G. Janssenswillen and B. Depaire, "bupar: Business process analysis in r." in *International Conference on Business Process Management - Demonstration track*, 2017.

[3] ——, *xesreadR: Read and Write XES Files*, 2019, R package version 0.2.3. [Online]. Available: https://CRAN.R-project.org/package=xesreadR

[4] W. van der Aalst, *Process mining: discovery, conformance and enhancement of business processes*. Heidelberg: Springer, 2011.

[5] N. Martin, G. Van Houdt, and G. Janssenswillen, "Towards more structured data quality assessment in the process mining field: the daqapo package," in *Proceedings of the European R Users Meeting 2020*, 2020.

[6] N. Martin, G. Van Houdt, and G. Janssenswillen, *daqapo: Data Quality Assessment for Process-Oriented Data*, 2020, R package version 0.3.0.

[7] A. J. M. M. Weijters and J. T. S. Ribeiro, "Flexible heuristics miner (FHM)," in *CIDM*. IEEE, 2011, pp. 310–317.

[8] A. Burattin and A. Sperduti, "Heuristics miner for time intervals," in *ESANN*, 2010.

[9] G. Janssenswillen, *propro: Build Probabilistic Process Models Using MCMC*, https://github.com/bupaverse/propro.

[10] G. Janssenswillen, B. Depaire, and F. Christel, "Enhancing discovered process models using bayesian inference and mcmc," in *Proceedings of the 2020 BPI Workshop*, 2020.

[11] L. Jooken and G. Janssenswillen, *collaborateR: Build Collaboration Graph Using Version Control System Logs*, R package version 0.1.0.

[12] L. Jooken, M. Creemers, and M. Jans, "Extracting a collaboration model from vcs logs based on process mining techniques," in *International Conference on Business Process Management*. Springer, 2019, pp. 212–223.

[13] C. W. Günther and W. M. Van Der Aalst, "Fuzzy mining–adaptive process simplification based on multi-perspective metrics," in *International conference on business process management*. Springer, 2007, pp. 328–343.

[14] M. Jans, P. Soffer, and T. Jouck, "Building a valuable event log for process mining: an experimental exploration of a guided process," *Enterprise Information Systems*, vol. 13, no. 5, pp. 601–630, 2019.

[15] A. Burattin, "Integrated, ubiquitous and collaborative process mining with chat bots." in *BPM (PhD/Demos)*, 2019, pp. 144–148.

[16] S. Shershakov, "Enhancing efficiency of process mining algorithms with a tailored library: Design principles and performance assessment."

[17] S. Kuehnel, S. T.-N. Trang, and S. Lindner, "Conceptualization, design, and implementation of econbpc–a software artifact for the economic analysis of business process compliance," in *International Conference on Conceptual Modeling*. Springer, 2019, pp. 378–386.

[18] M. Mesabbah and S. McKeever, "Presenting a hybrid processing mining framework for automated simulation model generation," in *2018 Winter Simulation Conference (WSC)*. IEEE, 2018, pp. 1370–1381.

[19] F. Mannhardt and A. D. Landmark, "Mining railway traffic control logs," *Transportation research procedia*, vol. 37, pp. 227–234, 2019.

[20] A. P. Kurniati, C. McInerney, K. Zucker, G. Hall, D. Hogg, and O. Johnson, "A multi-level approach for identifying process change in cancer pathways," in *International Conference on Business Process Management*. Springer, 2019, pp. 595–607.

[21] J. P. Salazar-Fernandez, M. Sepúlveda, and J. Munoz-Gama, "Influence of student diversity on educational trajectories in engineering high-failure rate courses that lead to late dropout," in *2019 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2019, pp. 607–616.

[22] D. Etinger, T. Orehovački, and S. Babić, "Applying process mining techniques to learning management systems for educational process model discovery and analysis," in *International Conference on Intelligent Human Systems Integration*. Springer, 2018, pp. 420–425.

[23] J. Saint, D. Gašević, W. Matcha, N. A. Uzir, and A. Pardo, "Combining analytic methods to unlock sequential and temporal patterns of self-regulated learning," in *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, 2020, pp. 402–411.

[24] D. Gašević, W. Matcha, J. Jovanović, A. Pardo, L.-A. Lim, S. Gentili *et al.*, "Discovering time management strategies in learning processes using process mining techniques," in *European Conference on Technology Enhanced Learning*. Springer, 2019, pp. 555–569.

[25] M. Tipirishetty, "Predictive process monitoring for lead-to-contract process optimization," 2016.

[26] B. A. Tama and M. Comuzzi, "An empirical comparison of classification techniques for next event prediction using business process event logs," *Expert Systems with Applications*, vol. 129, pp. 233–245, 2019.

[27] P. Delias and I. Kazanidis, "Exploiting higher-order dependencies for process analytics. the case for political events' analysis," *Kybernetes*, 2019.

[28] W. Ma, "Bias assessment and reduction in kernel smoothing," 2018.

[6] https://www.mmertens.eu/2020/06/process-mining-with-power-bi-and-r-visuals/

[7] https://www.linkedin.com/pulse/how-analyze-business-process-powerbi-using-r-visuals-peter-pensotti/?articleId=6631215429794836480

[8] https://stuifbergen.com/2018/08/analyse-web-site-click-paths-as-processes/