Application of process mining approach to the developmental process of the roundworm *C. elegans*

Trifon Chervenkov
Department of Medical Genetics
Medical University Varna
Varna, Bulgaria
ORCID [0000-0002-4964-1795]

Stoyan Pavlov
Department of Anatomy and Cell Biology
Medical University Varna
Varna, Bulgaria
ORCID [0000-0002-9322-2299]

Damyan Marinov
Department of Computer Science
Varna Free University
Varna, Bulgaria
ORCID [0000-0002-5179-8927]

Hristo Hristov
Department of Computer Science
Varna Free University
Varna, Bulgaria
ORCID [0000-0003-3696-439X]

Galina Momcheva
Department of Computer Science
Varna Free University
Varna, Bulgaria
ORCID [0000-0003-0726-2022]

***Abstract*—Process mining is an analytical approach which stems from and converges on data science and process modelling. Initially incepted to support business process management, however process mining approach is universal and applicable to other fields. It was already discerned that process mining techniques share similarities with such used in bioinformatics and that the emerging process mining discipline can benefit from applying techniques developed in computational biology [1]. Herein however, we demonstrate the reverse: that process mining can be applied for the study biological processes. As process mining operates on event logs in order to analyze a particular biological process it is necessary to transform the information for a sequence of biological events into an event log. For this study we applied process mining techniques to a developmental dataset from the lineage-resolved molecular atlas of the round worm *C. elegans* [2]. The single-cell temporal gene expression data was transformed into event log and analyzed with process mining tools. We show that application of process mining to biological processes is feasible, yet the presentation of the results with current tools is not suitable for the high information content of the particular biological process and this hampers further extraction of knowledge. We conclude that the application of process mining to biological processes would be beneficial for both fields.**

Keywords—process modelling, developmental biology, process mining, event log

I. INTRODUCTION

Process mining is a relatively young research discipline that sits between computa-tional intelligence and data mining on the one hand, and process modelling and anal-ysis on the other hand. The basic idea of process mining is to discover, monitor and improve real processes by extracting knowledge from event logs readily available in today's systems. Process mining includes process discovery, conformance checking, social network/organizational mining, automated construction of simulation models, model extension, model repair, case prediction, and history-based recommendations (IEEE Task Force on Process Mining). It was incepted in the field of business process modelling as a toolset to extract knowledge from business event logs and to optimize business processes. This approach, however, is universal as it is agnostic to the under-lying systems and is applicable to other fields, for instance it could be applied directly in healthcare if we describe patients as clients [3]. An originator of process mining is considered Wil Van der Aalst, a Dutch computer scientist and author of more than 400 books, articles, and publications, for his ground-breaking work on process mining [4]. Importantly, although process mining has a lot in common with data mining it is a different and separate approach as for instance data mining is data-centric and process mining is process-centric [4].

It was recognized that activities such as sequence analysis is fundamental and shared between process mining and bioinformatics and that both disciplines can benefit from cross-fertilization between the fields [1] and demonstrated that the emerging process mining discipline could benefit from techniques developed for bioinformatics. Here, we are proposing the reverse: to apply process mining techniques to biological processes in order to extract knowledge.

As the subject of process mining are time-based logs of a sequence of events best suited for process mining would be biological processes evolving in time. As such, we have selected for modelling the developmental process of the roundworm *C. elegans* as it is widely used as a model organism in developmental biology due to its relatively simple organization and short life cycle. For modelling, the publicly accessible dataset from the lineage-resolved molecular atlas of *C. elegans* was used [2], which contains single-cell RNA sequencing-derived time-resolved gene expression series. The associated dataset (GSE126954) contains a cell annotations list of 89,701 cells, including arbitrary embryonic time and gene expression data for all genes in the annotated cells. Additionally, the authors annotated cell lineages by confocal movies of marker genes and single-cell RNA sub-UMAP trajectories and calculated differential gene expression between pairs of cell lineages which is available as supplementary information [2]. The information supplied was sufficient to create event log for process mining, containing the minimum information of a case, event and timestamp.

The aim of the current work is to convert biological time-resolved gene expression data of *C. elegans* development to an event log and to analyze it by means of process mining.

II. METHODS

A. Conversion of developmental data to an event log

Useful for the event log conversion were supplementary tables S10 and S11 [2], containing differential gene expression between pairs of sister lineages (table S10) and pairs of parent-daughter lineages (table S11). The tables contain the names of the pairs of lineages under comparison, Gene ID from

978-1-6654-4581-8/22/$31.00

WormBase version WS260 and human-readable gene name from WormBase version WS260, expression level of the gene in both lineages, measured in transcripts per million (TPM), log2 TPM ratio be-tween lineages, p-value of the likelihood ratio test for differential gene expression between the lineages under comparison, p-value 95% confidence interval for the lineages under comparison, q-value (false detection rate) of the likelihood ratio test for differential gene expression between the lineages under comparison, q-value 95% confidence interval for the lineages under comparison and classification for gene expression change based on log2 TPM ratio between lineages. A detailed description of tables can be found as a supplementary information [2]. In these tables, the in-formation is aggregated and contains a comparison between groups of cells from a given lineage, rather than single cells as is in a cell annotation table. Due to this aggregation, embryonic time information was lost and was substituted with average embryo time of all annotated cells in a given cell lineage. In the source tables, the differential gene expression for each gene was categorized into expression gain and expression loss between pairs of lineages and this was used to create an event in the event log with four types of events for each gene for each lineage: expression gain vs sister, expression lost vs sister, expression gain vs parent and expression lost vs parent. In the final event log, each entry contained cell lineage, average embryo time of all cells for that lineage, gene id, gene name and one of the four gene expression events in comma-separated value (csv) format.

A detailed description of the supplementary tables S10 and S11 [2] transformations in Microsoft Access 2017 is as follows:

1. Table "cell_simple" is created from "cell_annotation", as the original data is grouped by the field "lineage", and the values of fields "embryotime", "rawembryo-time" and "Size_Factor" are averaged respectively in the following fields: "Avg_embryotime", "Avg_rawembryotime" and "Avg_Size_Factor". The fields "max_celltype" and "max_cellsubtype" are added, as informational fields, from the MAX values of the original fields "celltype" and "cellsubtype".

2. Table "gene_annotation" is not modified.

3. In table “table_s10” a new field is added “expression_gained_vs_sister”, with its values calculated with the following formula (similar to the field “expres-sion_gained_vs_parent”):

Set to TRUE if: TPM_95p_CI_lower_bound > 0 AND q_value_vs_sister < 0.05 AND ((sister_TPM_95p_CI_lower_bound == 0 AND log2_TPM_ratio_vs_sister >= 3.0) OR log2_TPM_ratio_vs_sister >= 4.0)

4. In table “table_s11” a new field is added - “expression_gained_vs_parent” - with values equal to the field “expression_gained_in_daughter”, and also a new field “expression_lost_vs_parent”, with values equal to the field “expres-sion_lost_in_daughter”.

5. The data from table "table_s10" is inserted to table "table_final_raw", as fol-lows:

a. Only where "expression_gained_vs_sister" = TRUE, the fields "lineage", "gene_id" and "expression_gained_vs_sister" are inserted into the fields with the same names. The fields "Avg_embryotime", "Avg_rawembryotime", "Avg_Size_Factor", "max_celltype" and "max_cellsubtype" are sourced from table "cell_simple", by join-ing by the field "lineage".

b. For the data where "expression_gained_vs_sister" = TRUE, the fields "sis-ter_lineage", "gene_id" and "expression_gained_vs_sister" are inserted into the fields "lineage", "gene_id" and "expression_lost_vs_sister" (i.e. the "gains" for the first sister are considered "losses" for the second sister). The fields "Avg_embryotime", "Avg_rawembryotime", "Avg_Size_Factor", "max_celltype" and "max_cellsubtype" are sourced from table "cell_simple", by joining by the field "sister_lineage".

c. The field "gene_name" is populated from table "gene_annotation", through joining by field "id" with field "gene_id".

6. The data from table "table_s11" is added into "table_final_raw", as follows:

a. Where "expression_gained_vs_parent" = TRUE, the fields "lineage", "gene_id" and "expression_gained_vs_parent" are inserted into the fields with the same names. The fields "Avg_embryotime", "Avg_rawembryotime", "Avg_Size_Factor", "max_celltype" and "max_cellsubtype" are sourced from table "cell_simple", by join-ing by the field "lineage".

b. For the data where "expression_lost_vs_parent" = TRUE, the fields "lineage", "gene_id" and "expression_lost_vs_parent" are inserted into the fields with the same names. The fields "Avg_embryotime", "Avg_rawembryotime", "Avg_Size_Factor", "max_celltype" and "max_cellsubtype" are sourced from table "cell_simple", by join-ing by the field "lineage".

c. The field "gene_name" is populated from table "gene_annotation", through joining by field "id" with field "gene_id".

7. Table "table_final" is created from table "table_final_raw", by aggregating the data and grouping by the field "lineage".

B. Process mining of the event log

Process mining of the created event log was preformed via ProM 6.10 software package [5]. First, the event log in csv format was imported via the "Convert CSV to XES" ProM plugin with selecting Case Column: lineage, Event Column: gene name and event and Start Time: embryo time in "sss" format. Combining gene name and event type as an integral event allowed the gene expression event to be indivisible from the gene name.

The converted log was further analysed with "ILP-Based Process Discovery (Ex-press)" plugin with default parameters. The result was presented as a Petri net and was exported as a Petri Net Markup Language (PNML) file. It was further visualized with Graphviz Petri net visualization plugin and was exported in Scalable Vector Graphics (SVG) format.

III. Results and Discussion

The result is a Petri net with high complexity and because of this not very comprehensible (“Fig. 1”) [5]. Selecting the event column to be only gene expression change, excluding gene name from the event, caused severe underfitting and did not produce meaningful results (data not shown).

Fig. 1. Petri net diagram of the ILP-Based Process Discovery analysis of the generated event log.

The results are graphical only and not easy to analyze further as they are not much interactive: there is no search tool interactive zooming. These limitations were already pointed by Bose et al. [1] and either software tools improvement is needed or the resulting Petri net have to be exported and analysed by external instruments like Cytoscape [6]. The latter is not a trivial task: Cytoscape can process Petri nets, but the software does not natively support the PNML file format.

Nevertheless, in "Fig. 2" can be seen a sector of the Petri net where four genes dyf-1, dyf-2, dyf-3 and dyf-5 are sequentially linked together, and further investigation shows that the function of all is associated with the same structure: cilium. This graphical association may be reflecting a biologically relevant interaction.

Similarly, in "Fig. 3" can be seen a sector of the Petri net where two genes daf-7 and daf-10 are sequentially linked together, and both genes are involved in dauer formation. Again, this may be reflecting a biologically relevant interaction.

Maybe the most interesting relation we were able to find is presented on "Fig. 4": atg-10 and tbc-14 genes. Atg-10 is related to autophagy and the human orthologs of tbc-14 - TBC1D15 and TBC1D17 are also involved in this biological process [7]. This finding may be the most biologically relevant interaction we were able to observe graphically.

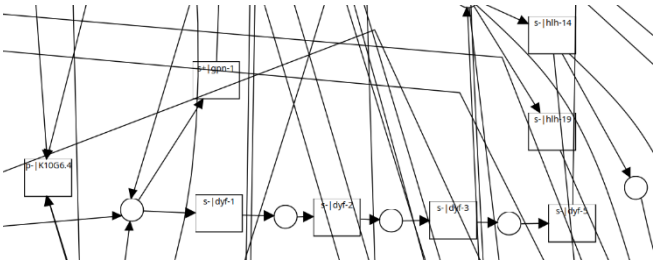

Fig. 2. Section of the Petri net diagram, linking the loss of expression in sister lineage of 4 related genes: *dyf-1*, *dyf-2*, *dyf-3* and *dyf-5*.

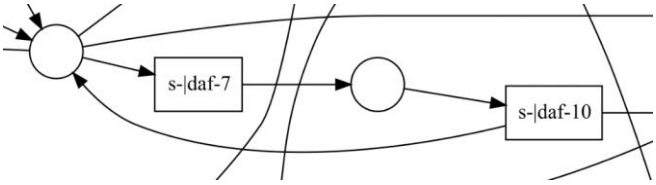

Fig. 3. Section of the Petri net diagram, linking the loss of expression in sister lineage of two genes: daf-7 and daf-10.

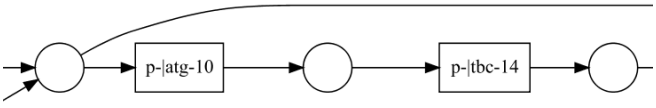

Fig. 4. Section of the Petri net diagram, linking the loss of expression in parent lineage of two genes: atg-10 and tbc-14.

IV. Conclusions

To our knowledge process mining was applied to biological processes for the first time. In this preliminary work, we can conclude that its translation is feasible. However, the current tools are not convenient to use and are limiting the full potential of the application of process mining on biological processes. A critical limitation of the tools used was the non-interactive visualization which hampered further biological analysis, and this highlights the importance of visual analysis. Another possible limitation of the tool used was the necessity of case selection: an algorithm was proposed to operate on event logs without case id – correlation miner [8] – and this may be more informative.

References

[1] R. P. J. C. Bose and W. M. P. van der Aalst, "When Process Mining Meets Bioinformatics," Berlin, Heidelberg, 2012, pp. 202-217.

[2] J. S. Packer, Q. Zhu, C. Huynh, P. Sivaramakrishnan, E. Preston, H. Dueck, *et al.*, "A lineage-resolved molecular atlas of C. elegans embryogenesis at single-cell resolution," *Science*, vol. 365, Sep 20 2019.

[3] R. S. Mans, W. M. P. van der Aalst, and R. J. B. Vanwersch, *Process Mining in Healthcare: Evaluating and Exploiting Operational Healthcare Processes*: Springer International Publishing, 2015.

[4] W. M. P. van der Aalst, *Process Mining: Data Science in Action*: Springer Berlin Heidelberg, 2016.

[5] E. Verbeek, J. C. A. M. Buijs, B. F. v. Dongen, and W. M. P. van der Aalst, "ProM 6: The Process Mining Toolkit," *European Journal of Operational Research*, 2010.

[6] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, *et al.*, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome Res*, vol. 13, pp. 2498-504, Nov 2003.

[7] T. Itoh and M. Fukuda, "Chapter 6 - Roles of Rab-GAPs in Regulating Autophagy," in *Autophagy: Cancer, Other Pathologies, Inflammation, Immunity, Infection, and Aging*, M. A. Hayat, Ed., ed: Academic Press, 2017, pp. 143-157.

[8] S. Pourmirza, R. Dijkman, and P. Grefen, "Correlation Miner: Mining Business Process Models and Event Correlations Without Case Identifiers," *International Journal of Cooperative Information Systems*, vol. 26, p. 1742002, 2017.