Visualisation à l'Échelle de Séquences d'Évènements Issus de Dossiers Médicaux Électroniques

Visualization of Electronic Health Record Sequences at Scale

Ambre Assor, Mickaël Sereno, Jean-Daniel Fekete

English Abstract—We present ParcoursVis, a Progressive Visual Analytics tool designed to explore Electronic Health Record sequences of patients at scale. Existing tools process and aggregate the whole dataset upfront before showing the visualization, taking a time proportional to the data size. Therefore, to remain interactive, existing tools are limited to data sizes that can be processed in under a few seconds to meet the latency constraints of human attention. To overcome this limitation and scale to larger sizes, ParcoursVis relies on a progressive algorithm that quickly shows an approximate initial result of the aggregation, visualized as an Icicle tree, and improves it iteratively, updating the visualization until the whole computation is done. With its architecture, ParcoursVis remains interactive while visualizing the sequences of tens of millions of patients, each described with thousands of events; three orders of magnitude more than similar systems. Managing large datasets allows for exploring rare medical conditions or unexpected patient pathways, contributing to improving treatments. We describe the algorithms we use and our evaluation concerning their scalability, convergence, and stability. We also report on a set of guidelines to support visualization designers in developing scalable progressive systems. ParcoursVis already allows practitioners to perform analyses on two real large medical datasets. Our prototype is open-source.

1 PARCOURVIS

As datasets continue to increase in size and complexity, traditional visualization techniques struggle to maintain interactivity and responsiveness, thereby hindering effective exploratory data analysis. Addressing the scalability challenge necessitates progress in algorithmic implementations and the development of optimized data structures within visualization systems, but more importantly, a change in programming paradigm, using progressive visual analytics (PVA) to decouple data size and algorithm complexity from latency [2].

The scalability challenge is particularly acute in healthcare, where Electronic Health Records (EHRs) contain vast amounts of patient information. For instance, the French National Health Database [7] holds decades of health data on citizens, encompassing records of drug purchases, medical diagnoses, and hospital treatments. Such data have the potential to advance healthcare by enabling a better understanding of treatment practices and patient outcomes. As an example, by reconstructing and visualizing temporal event sequences from drug purchase records, the examination of patients' treatment journeys over time is facilitated. This approach aligns with the goal of medical professionals to assess whether patients' treatments adhere to national guidelines and, if deviations occur, to understand the reasons. This is key to providing patients with the highest quality of care through evidence-based and safe recommendations. Yet, the volume of such datasets necessitates innovative visualization approaches capable of handling large-scale medical data while preserving analytical efficiency and usability. In this context, we developed ParcoursVis, a PVA tool created in collaboration with medical institutions to visualize patient care pathways derived from EHRs.

Existing work has laid a foundation for similar visualization tools aimed at healthcare analysis (e.g., [1], [4], [6]). For instance, EventFlow [5] aggregates event sequences into a *prefix tree* and visualizes them as an *Icicle Tree*; a visualization that has proven effective in identifying optimal drug usage patterns. We, therefore, build on the EventFlow visualization technique for ParcoursVis, along with several improvements.

However, tools like EventFlow have generally been applied to relatively small datasets, sometimes encompassing fewer than 100 patients. According to our experiments, EventFlow is limited to about 20k patients. While such tools have offered valuable initial insights, they fall short in scalability, which has been reported as a main challenge for EHR visualization [10]. In practice, regional and national EHR databases present data at a far greater scale, often involving millions of patients. To our knowledge, no

- *Ambre Assor: Université Paris-Saclay, CNRS, and Inria E-mail: ambre.assor@inria.fr.*
- *Mickaël Sereno: Université Paris-Saclay, CNRS, and Inria E-mail: mickael.sereno@inria.fr.*
- *Jean-Daniel Fekete: Université Paris-Saclay, CNRS, and Inria E-mail: jean-daniel.fekete@inria.fr.*

existing tool enables the management and interactive analysis of EHR datasets exceeding 100k event sequences [12], [11], [5], [10], which limits the scope of exploration. This constraint impedes the ability to gain a more comprehensive view of patient pathways, identify patterns and trends that may not be visible in smaller or sampled datasets, and discover rare patterns and outliers. With ParcoursVis, we allow visualizing temporal event sequences of more than 100M patients, potentially supporting the largest EHRs.

Large-scale interactive data processing poses significant challenges, particularly in terms of response times. To accommodate the need for a responsive system that supports health analysts in conducting explorations on large-scale EHR data, we rely on a Progressive Visual Analytics architecture [8], [2]. Our progressive approach provides a quick, approximate, yet useful preview of the data early on to maintain user attention within human latency limits and quick updates converging within seconds. Indeed, results from research in visualization show that user attention declines after 500 ms, and after five to ten seconds, users tend to abandon tasks or lose focus [13], [3]. After the initial preview, our algorithm continues processing the dataset, updating the visualization every second or so until it is fully processed.

Yet, the progressive updates of the aggregated tree can cause visual instability, as the nodes are laid out at each update. The shifting of nodes can disrupt the viewer's mental map and create distracting visual flickers. Therefore, our approach may have an impact on the usability of the tool. To mitigate this issue, we introduce a *sorting with hysteresis* algorithm that limits this flicker during the progressive rendering.

According to the survey of Ulmer et al. [9], we describe the first progressive tree visualization algorithm. We outline its specific issues related to stability and uncertainty.

ParcoursVis improves the scalability of aggregated EHR event sequence visualization by at least three orders of magnitude compared to state-of-the-art systems. Its source code and evaluation scripts are available at https://gitlab.inria.fr/aviz/parcoursvis/.

2 Results

Comparison with existing systems We tried the EventFlow [5] program with a subset of our dataset containing 10k, 20k, and 50k synthetic patients, comparing it to ParcoursVis. EventFlow struggles with datasets greater than 10K patients, showing delays up to 40s for loading times (for 50K patients) and up to 410s for visualization; showing back or merging events take longer than 10s, and memory usage increases significantly (13GB for 10k, 28GB for 50k). ParcoursVis supports datasets 100–1,000 times larger within a 2-second latency, even for 20M patients.

Scalability We evaluated the scalability of our algorithm by processing a synthetic dataset of 10M patients, measuring computation times for both progressive and non-progressive environments. Using up to six threads, we achieved processing speeds from 2.67M patients per second (single thread) to 13M patients per second (six threads). The results show that computation time is linear with respect to the number of low-level events and inversely proportional to the number of threads. Tree merge time is negligible, and the algorithm scales efficiently with increasing chunk sizes[1] and threads.

Stability We evaluated the stability and convergence of two sorting algorithms: *Hysteresis Sort* and regular sort (*RSort*), used in ParcoursVis for visualizing aggregated trees. *Hysteresis Sort*, with a small inertia, prevents visual instability by maintaining node order when frequencies are close. Our results show that *Hysteresis Sort* stabilizes nodes faster than *RSort*, particularly for deeper tree levels. The stability of nodes is influenced by the minimum node size and chunk size, with larger nodes stabilizing sooner. *Hysteresis Sort* improves stability, especially for low-frequency nodes, compared to *RSort*.

Usability We provided ParcoursVis to referring doctors to explore a non-cancerous prostate adenoma dataset. We received positive feedback. We also used ParcourVis for visualizing patient pathways in the 16 Public Emergency Departments (EDs) in Paris. We interviewed, in an ongoing usability study, four professionals interested in analyzing patients' pathways (an emergency physician highly motivated to find more effective organizational approaches, a computer scientist aiming at achieving high-fidelity simulation of patients' flow in EDs, and two clinical researchers working on EDs data). None of our participants mentioned any latency issues, not even performance issues. They were able to focus on their professional questions.

3 Conclusion

We presented ParcoursVis, a PVA tool we designed to explore patients' care pathways at scale. Regarding our results on scalability, stability, and usability, ParcoursVis can be used effectively to explore patients' treatments and conditions aggregated at the scale of the largest countries to improve public health based on data. Each new study requires new features, and we are planning to extend ParcoursVis to make it more generic and applicable to new use cases without programming. We also plan to support datasets updated continuously. By providing our system in open-source, we want to push this research field to a wider audience relying on PVA to provide scalability and improve public health.

1. In a progressive paradigm, data is processed chunk by chunk compared to treating the whole dataset at once.

References

[1] M. V. Bjarnadóttir, S. Malik, E. Onukwugha, T. Gooden, and C. Plaisant. Understanding Adherence and Prescription Patterns Using Large-Scale Claims Data. *PharmacoEconomics*, 34:169–179, 2016. 1

[2] J.-D. Fekete, D. Fisher, and M. Sedlmair. *Progressive Data Analysis: Roadmap and Research Agenda*. Eurographics, Nov. 2024. 1, 2

[3] Z. Liu and J. Heer. The Effects of Interactive Latency on Exploratory Visual Analysis. *IEEE Trans. Vis. Comput. Graph.*, 20(12):2122–2131, 2014. 2

[4] T. E. Meyer, M. Monroe, C. Plaisant, R. Lan, K. Wongsuphasawat, T. S. Coster, S. Gold, J. Millstein, and B. Shneiderman. Visualizing Patterns of Drug Prescriptions with EventFlow: A Pilot Study of Asthma Medications in the Military Health System. In *Proc. VAHC*, pages 55–58, New York, 2013. ACM. 1

[5] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal Event Sequence Simplification. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2227–2236, 2013. 1, 2

[6] M. Ozkaynak, O. Dziadkowiec, R. Mistry, T. Callahan, Z. He, S. Deakyne, and E. Tham. Characterizing workflow for pediatric asthma patients in emergency departments using electronic health records. *Journal of Biomedical Informatics*, 57:386–398, 2015. 1

[7] Système National des Données de Santé (SNDS). https://www.snds.gouv.fr/. Accessed: 2023-02-01. 1

[8] C. D. Stolper, A. Perer, and D. Gotz. Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1653–1662, 2014. 2

[9] A. Ulmer, M. Angelini, J.-D. Fekete, J. Kohlhammer, and T. May. A Survey on Progressive Visualization. *IEEE Trans. Vis. Comput. Graph.*, 30(9):6447–6467, 2024. 2

[10] Q.-R. Wang and R. S. Laramee. EHR STAR: The State-Of-the-Art in Interactive EHR Visualization. *Comput. Graph. Forum*, 41(1):69–105, 2022. 1, 2

[11] K. Wongsuphasawat and D. Gotz. Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2659–2668, 2012. 2

[12] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman. LifeFlow: Visualizing an Overview of Event Sequences. In *Proc. CHI*, pages 1747—-1756, New York, 2011. ACM. 2

[13] E. Zgraggen, A. Galakatos, A. Crotty, J.-D. Fekete, and T. Kraska. How Progressive Visualizations Affect Exploratory Analysis. *IEEE Trans. Vis. Comput. Graph.*, 23(8):1977–1987, 2017. 2