ProChess: Customizable Event Log Generation from Chess Game Data

Sabine Nagel*

University of Koblenz, Universitätsstr. 1, 56070, Koblenz, Germany

Abstract

We present ProChess, a browser-based tool for transforming chess game data into customizable XES event logs for process mining. Chess games are fully observable traces of real human decision-making and freely available at scale through platforms such as Lichess and Chess.com. ProChess allows users to upload PGN files, slice and filter datasets by various properties, such as player, opening, and time control, and configure activity labels at varying levels of granularity. The generated logs are compatible with standard process mining tools and enable novel benchmarks, process discovery and conformance checking experiments, and engaging educational use cases.

Keywords

Process Mining, Process Discovery, Event Log, Chess

1. Introduction and Related Work

Process mining techniques rely on event logs as their fundamental input for process discovery, conformance checking, and performance analysis. A well-known limitation of the field is the scarcity of publicly available, high-quality real-world event data. While benchmark collections such as the BPI Challenge logs are widely used, obtaining real-world data from companies has always been difficult [1], and the available logs cover only a narrow range of process types. Synthetic event logs allow for controlled evaluation with known ground truth, but by design do not capture the full complexity and variability of real human behavior [1].

Sports data has emerged as a promising alternative source for process mining. Kröckel and Bodendorf [2] and Chan et al. [3] have shown that structured competitive behavior from football can be productively analyzed with process mining, as it yields tactical insights not visible through traditional statistics. However, professional sports data is largely proprietary, covers only specific matches or tournaments, and tends to exhibit high variability that complicates process discovery.

Chess game data avoids all of these limitations and offers additional unique advantages. Each game is a complete, fully observable trace of real human decision-making under time pressure and strategic uncertainty, with no missing or ambiguous events. Unlike football or business process data, chess additionally provides full observability of alternative actions at each decision point, since all legal moves are computable from the board state, which enables translucent forms of process analysis that are simply not possible with conventional logs. Large-scale datasets are freely and permanently available through platforms such as Lichess and Chess.com. Ferilli and Angelastro [4] have already shown that meaningful patterns can be extracted from chess games through process mining, which confirms that the domain is genuinely promising.

We present ProChess, a browser-based tool that views chess as a process and transforms raw chess game data into customizable, standards-compliant XES event logs. Users can upload PGN files, slice and filter datasets by various criteria and configure activity labels at varying levels of granularity. As a result, chess data not only provides a novel class of freely available, real-world benchmarks, but also

Proceedings of the BPM 2026 Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum (BPM 2026) , Toronto, Canada, September 27th to October 2nd, 2026.

*Corresponding author.
snagel@uni-koblenz.de (S. Nagel)
0000-0003-4838-8246 (S. Nagel)

enables the application and communication of process mining techniques to a new field and audience. Additionally, the use of chess as a familiar and engaging domain might be a promising approach to support process mining education. Analyzing well-known games or students' own matches has the potential to be an engaging entry point into process discovery and conformance checking, as it makes abstract process mining concepts more accessible to learners.

The remainder of this paper is structured as follows. Section 2 introduces the relevant chess concepts and available attributes in chess game data. Section 3 describes ProChess and its four phases in detail. We discuss usage and maturity in Section 4 and conclude in Section 5.

2. Preliminaries

A chess game is played between two sides, *White* and *Black*, who alternate making moves on an 8×8 board. Each side controls six piece types: pawn (P), knight (N), bishop (B), rook (R), queen (Q), and king (K), each with fixed movement rules. Squares are identified by a file letter (a–h) and a rank number (1–8). Moves are recorded in *Standard Algebraic Notation* (SAN), encoding the piece type, destination square, and any special flags such as captures (x), checks (+), or checkmate (#); for example, `Bxe5+` denotes a bishop capturing on square e5 with check. Special moves include *castling* (king and rook swap, written `O-O` or `O-O-O`), *en passant* (a pawn capturing a passing pawn), and *promotion* (a pawn reaching the last rank and converting to a stronger piece, e.g., written `e8=Q`).

Games are distributed in *Portable Game Notation* (PGN), a plain-text format with header fields (players, Elo ratings, event, date, result, time control) followed by the move sequence. When recorded by electronic boards, move comments include clock annotations (`%clk H:MM:SS`) giving the time remaining on each player's clock after every move. Time controls specify how much time each player receives (e.g., 90 minutes for the first 40 moves with a 30-second increment per move, followed by 30 additional minutes). Openings are classified by *ECO codes* (Encyclopaedia of Chess Openings), a system of roughly 500 named opening lines. The large number of codes reflects the combinatorial diversity of early move sequences, where even small differences in the first few moves lead to strategically distinct positions.

Listing 1 shows a PGN excerpt for the first five moves of a Carlsen vs. Nepomniachtchi game from the FIDE World Chess Championship 2021.

Listing 1: PGN excerpt

```
[Event "2021 FIDE World Chess Championship"]
[Date "2021.12.03"]  [Round "6"]
[White "Carlsen, Magnus"]  [Black "Nepomniachtchi, Ian"]
[WhiteElo "2855"]  [BlackElo "2782"]
[TimeControl "5400+30"]  [Result "1-0"]

1. d4 {[%clk 1:59:57]} Nf6 {[%clk 1:59:51]} 2. Nf3 {[%clk 1:59:53]} d5 {[%clk 1:59:35]} 3. g3
{[%clk 1:59:45]} e6 {[%clk 1:55:49]} 4. Bg2 {[%clk 1:59:38]} Be7 {[%clk 1:55:43]} 5. O-O {[%clk
1:59:32]} O-O {[%clk 1:55:30]}
```

3. Tool Description

ProChess is implemented as a browser-based web application built with Python and Flask. It is organized into four phases: PGN upload and parsing, slicing, filtering, and event log export, each accessible via a dedicated step in the interface. Additionally, the list of games based on set slices and filters can be viewed in the game explorer at any point in time. Users can freely navigate between phases, and all settings are preserved across the session. All configuration steps provide a selection of presets, so users can generate a usable event log with minimal effort and without deep chess knowledge. At the same time, each phase exposes fine-grained options for users who want full control over the dataset and log structure.

3.1. PGN Upload and Parsing

Users begin by uploading one or more PGN files, which can be obtained from platforms such as Chess.com or Lichess as well as a wide variety of game databases. During upload, ProChess fully parses every game before the user can continue. During parsing using the python-chess library[1], ProChess extracts both standard header fields (player names, Elo ratings, event, site, date, result, time control) (cf. Listing 1) and move-level properties derived by replaying each game on a board representation, such as the moving and captured pieces, source and target squares, and structural move flags (capture, check, checkmate, castling, en passant, promotion). Clock annotations are recorded by electronic boards and became widespread in professional play only from roughly 2015 onward. When present, per-move time usage and cumulative elapsed time are computed; otherwise, move order is used to create artificial timestamps, which ensures that a meaningful event ordering is always available. The time control category (bullet, blitz, rapid, classical) is derived from the time control header or estimated from the first observed clock time, if no time control header is present. Openings are identified by matching the move sequence against an ECO code catalog, and the game's termination reason (e.g., checkmate, resignation, insufficient material) is inferred from the final board state, the recorded result, and, where available, the header and clock information.

The remaining, computationally expensive move property is whether a move creates a direct or discovered attack on an opposing piece. Rather than delaying the end of the import phase, ProChess starts computing these attack flags for all games in a background thread as soon as the import step finishes, while the user is already free to slice, filter, and explore the dataset.

3.2. Slicing

Before narrowing the dataset with filters, users can choose how the eventual event log should be split into files. By default, ProChess produces a single event log containing all games in scope. Alternatively, users can select a sliced export, which partitions the games in scope along one dimension and produces one XES file per value of that dimension, bundled into a single archive. Available slicing dimensions include player, opening, game result, player title (e.g., Grandmaster), and custom Elo ranges defined by the user. Slicing is independent of and applied on top of the filters defined in the next phase, so a single filtered dataset can, for example, be exported as one file per opening for direct comparison.

3.3. Filtering and Determining Focus of Analysis

Users can now define the subset of games and moves to include in the analysis. Filters cover player selection, opening, piece color, game result and termination, player and opponent Elo range, player title, date, time control category, game length, variant (e.g., excluding Chess960), and whether clock timestamps are required. A live game count updates as filters are adjusted, and a warning is shown when the selection falls below a meaningful threshold. In that case, users can return to the upload phase to add more data, or switch to the Explorer to inspect the current selection and decide whether to relax the filters. When a single player or color is selected as the focus, only moves by the relevant side are included as events.

3.4. Game Explorer

The Explorer lists all games that match the defined focus. Clicking on a game opens a detailed view of that game, including a full move table and a chess board to replay the game for a more detailed exploration. This allows verifying that the current slice and filter configuration captures the intended data before proceeding to export.

[1]https://pypi.org/project/chess/

3.5. Event Log Definition and Export

The export phase allows users to configure and generate an XES event log compatible with standard process mining tools.

Users can select which game-level and move-level attributes to include. Game-level (case) attributes include player names, Elo ratings, event, site, date, result, opening, time control, and termination. Move-level (event) attributes include piece type, source and target square, tactical move type, SAN notation, clock time, per-move time, and cumulative elapsed time.

The central configuration decision is how to define the activity for each event. In addition to using the original SAN notation (which can yield upward of 40 000 theoretically distinct activities and is, therefore, often not a sensible choice for process mining), users can freely combine five move properties to construct the activity label: side (White or Black), piece type, source square, target square, and tactical move type. Such combinations yield far fewer, more analysis-friendly activities. For example, selecting piece type and target square produces activities such as *Pawn e4* or *Knight f3*, while adding move type yields *Pawn e4 capture* or *Knight f3 check*. To support an informed choice, ProChess computes the exact number of distinct activity labels for every possible property combination as filters are adjusted. A live preview table shows up to ten representative event rows for the current configuration, so the user knows what to expect before exporting.

Once the configuration is complete, ProChess generates the event log using PM4Py [5]. Each game becomes a trace with game-level case attributes. Each move becomes an event with `concept:name` set to the configured activity label and `time:timestamp` derived from cumulative elapsed clock time where available, or estimated from move order otherwise.

Table 1 shows the event log produced from the PGN excerpt in Listing 1 using *Piece + Target square* as the activity label, with a selection of additional game and event attributes.

Table 1
Example event log for Listing 1 (activity = Piece + Target square).

ID	Activity	SAN	Elapsed	Player	Elo	Side	Piece	From	To	Type
1	Pawn d4	d4	00:00:33	Carlsen	2855	White	Pawn	d2	d4	regular
1	Knight f6	Nf6	00:01:12	Nepomniachtchi	2782	Black	Knight	g8	f6	regular
1	Knight f3	Nf3	00:01:46	Carlsen	2855	White	Knight	g1	f3	regular
1	Pawn d5	d5	00:02:32	Nepomniachtchi	2782	Black	Pawn	d7	d5	regular
1	Pawn g3	g3	00:03:10	Carlsen	2855	White	Pawn	g2	g3	regular
1	Pawn e6	e6	00:07:26	Nepomniachtchi	2782	Black	Pawn	e7	e6	regular
1	Bishop g2	Bg2	00:08:03	Carlsen	2855	White	Bishop	f1	g2	regular
1	Bishop e7	Be7	00:08:39	Nepomniachtchi	2782	Black	Bishop	f8	e7	regular
1	King g1	O-O	00:09:15	Carlsen	2855	White	King	e1	g1	castling
1	King g8	O-O	00:09:58	Nepomniachtchi	2782	Black	King	e8	g8	castling

4. Usage and Maturity

ProChess is available for direct use in the browser without installation[2]. Several PGN datasets are pre-loaded, so users can begin exploring immediately. Users may also upload their own PGN files, for example games from their personal Chess.com or Lichess account, to analyze their own play or that of a specific player or opening repertoire. A screencast[3] demonstrating the full workflow is also available.

ProChess handles datasets of varying sizes efficiently. As described in Section 3, only the import step (header and move parsing, opening and termination detection) blocks the user, while the computationally more expensive attack-flag computation runs in a background thread in parallel with slicing, filtering, and exploration, and is only awaited if still incomplete when it is actually needed. As a result, slicing, filtering, and exploring remain responsive, even for large PGN collections.

[2]https://uni-ko.de/prochess
[3]https://uni-ko.de/prochess-screencast

5. Conclusion

We presented ProChess, a browser-based tool for transforming chess game data into customizable, standards-compliant XES event logs. The tool directly addresses a longstanding limitation of the process mining field, which is the scarcity of freely available, high-quality real-world event data. Chess games are fully observable, richly annotated traces of real human decision-making under time pressure, available at massive scale without institutional access barriers. ProChess makes this data accessible to the process mining community through flexible slicing and filtering, configurable activity granularity, and seamless XES export, which lowers the barrier for researchers and educators alike.

Beyond complementing existing benchmarks, chess data offers properties such as genuine human behavior combined with full observability and ground-truth knowledge of all possible actions at each decision point. Procedural process discovery can reveal typical move sequences and opening repertoires as process models to make implicit strategic patterns explicit and human-readable. Declarative process mining is particularly well-suited to chess, where play is governed by constraints rather than a fixed procedure, and can uncover rules and regularities that distinguish playing styles, skill levels, or game outcomes. Conformance checking against a discovered or hand-crafted model can identify deviations from expected play and flag unusual or suboptimal decisions in a principled way. More broadly, ProChess enables the application of the full BPM technology stack to a universally understood domain, which also makes it a promising tool to support process mining education.

Future work includes integration with Stockfish for engine-based move evaluation, which would enrich each event with attributes such as centipawn loss and move quality classification to enable deeper performance analyses. Additionally, we plan on extending ProChess with the possibility of calculating all legal alternatives for each move. This last property, unique to rule-based domains like chess, opens the door to counterfactual-aware process analysis by asking not only what players did but what they could have done instead. Furthermore, we aim to not only allow users to export flattened event logs but also extend ProChess with capabilities for object-centric event log generation, as chess is a naturally multi-object domain in which games, players, and individual pieces interact simultaneously. Modeling these as distinct object types would allow, for example, tracing the full lifecycle of a single piece across a game while preserving its relationships to other pieces and to the overall game outcome, something that a single case notion cannot capture without loss of information.

Declaration on Generative AI

During the preparation of this work, the author used Sonnet 4.6 for grammar and spelling check, as well as error handling during implementation.

References

[1] B. F. van Dongen, E. Verbeek, Challenges and contests in process mining, in: Mining a Scientist's Process: Essays Dedicated to Wil van der Aalst on the Occasion of His 60th Birthday, Springer, 2026, pp. 333–348.

[2] P. Kröckel, F. Bodendorf, Process mining of football event data: a novel approach for tactical insights into the game, Frontiers in artificial intelligence 3 (2020) 47.

[3] V. Chan, L. Ebert, P.-J. Hillmann, C. Rubensson, S. A. Fahrenkrog-Petersen, J. Mendling, Transforming football data into object-centric event logs with spatial context information, in: International Conference on Business Process Management, Springer, 2025, pp. 467–478.

[4] S. Ferilli, S. Angelastro, Mining chess playing as a complex process, in: International Workshop on New Frontiers in Mining Complex Patterns, Springer, 2016, pp. 248–262.

[5] A. Berti, S. van Zelst, D. Schuster, Pm4py: A process mining library for python, Software Impacts 17 (2023) 100556.