The Journal of Open Source Software

DataBallPy: Load, Synchronise, and Analyse your Soccer Data

Oonk [1], **Grob** [2], and **Kempe** [1,3]

1 Department of Sports Sciences, University of Groningen, the Netherlands **2** Independent Researcher, the Netherlands **3** Centre for Sport Science and University Sports, University of Vienna, Austria

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 28 January 2026
Published: unpublished

Summary

Over the last decade, there has been a growing interest in soccer analytics from different backgrounds and for different use cases. Examplary use cases are: first, practical decision making and benchmarking of players based on aggregated metrics such as pass success percentage and expected goals (xG) (Goes, Meerhoff, et al., 2020). Second, using internal and external load metrics for training periodization and injury predictions (Hader et al., 2019). Third, basic behavioural science soccer with a focus on group and subgroup behaviour(Goes, Brink, et al., 2020). The interest in soccer analysis has also increased since data has become more openly available (Bassek et al., 2025). However, a key challenge is that every data provider uses their own data format, which makes it hard to compare and switch between different providers and create large datasets that encompass different leagues and competitions. Currently, open-source packages like Kloppy try to overcome this challenge by providing a uniform data format. Similarly, the scientific side proposes a common data format for soccer game data (Anzer et al., 2025). While Kloppy focuses primarily on parsing soccer data, Floodlight (Raabe et al., 2022) delivers a framework for physical analysis of team sports, and mplsoccer is widely utilized for visualising soccer data.

Lately, there has been a growing interest in combining event and tracking data for contextualised tactical analysis of soccer games. This provides the possibility to not only know that a pass happened at a specific moment in the match (event data) but also what the defensive structure was during this pass (Forcher et al., 2022; Herold et al., 2022), and what other passing options were available at this moment (tracking data) (Spearman et al., 2017). Contextual analysis goes beyond aggregated metrics and provides the ability to do quantitative analysis of single moments or specific phases in the game (Jerome et al., 2024; Oonk, Buurke, et al., 2025). Merging tracking and event data is a key challenge for contextualised analysis of soccer games. `DataBallPy` is an open source python package for contextual analysis of soccer games because (1) it uses a standardized data format for both event and tracking data, (2) it provides a framework where all data of a game is bundled, instead of considered as seperate data objects, (3) it includes a high quality and learning free synchronsiation algorithm that works on any combination of tracking and event data providers, and (4) it has integrated multiple practical and scientific features within the package that allow for efficient computation with minimal user input.

Statement of need

Modern soccer analytics increasingly rely on both event data and tracking data for a comprehensive analysis. Event data captures specific information about events (e.g., passes and shots) like their location, success, start location, and the athlete involved in the action. This information on itself is primarily aggregated for tactical game and player analysis

(Goes, Meerhoff, et al., 2020) but is also widely used in scouting because of the low cost and widespread availability of the data (Arem et al., 2025). Tracking data, on the other hand, captures spatiotemporal information of all athletes and the ball at frequencies ranging between 10 and 25 Hz (Linke et al., 2020). This data is primarily used to quantify physical performance, but also for the detection of dynamic formation (Sotudeh, 2025), detection of events (Vidal-Codina et al., 2022), detection of game phases (Bauer et al., 2023), space occupation (Rein et al., 2017; Spearman et al., 2017), and quantification of dangerousity (Link et al., 2016).

The currently avaiable packages allow for parsing (Kloppy) and analysis (Raabe et al., 2022) of either data stream independently. However, there has been a growing interest in combining event and tracking data to enrich event information with spatiotemporal context. This added context provides insights and nuances, primarily on a tactical level, that neither event nor tracking data can provide independently. For example, shot events are enriched with information about defensive and keeper positioning to create better expected goals models (Anzer & Bauer, 2021), passes are evaluated by making risk reward assessments of all possible passing options (Goes et al., 2021), determinants of successful 1v1 actions are modelled from spatiotemporal features (Oonk, Buurke, et al., 2025), and the spatiotemporal context of events is used to predict dangerousity of a game state (Fernández et al., 2021). A contextual analysis requires a proper synchronisation of event and tracking data, and a convenient data structure for further analysis. Current packages either have a separation between event and tracking data with limited options to combine them (Raabe et al., 2022), or focus only on the synchronistation approach, limiting the convenient data structure to start your analysis after merging the data streams (Kim et al., 2025; Roy et al., 2024)

`DataBallPy` addresses this gap by combining all game-related data in a standardized `Game` object. The `Game` object includes event, tracking, and metadata. The primary feature of `DataBallPy` is the robust and efficient synchronistation between event and tracking data. Although event and tracking data often both provide timestamps, their alignment has shown to be extremely poor with reported errors of 1.82 (+-4.06) seconds (Anzer & Bauer, 2021). Especially, the random error is concerning since it does not allow for easy correction, and within 4 seconds, the game might have evolved to an entirely different situation. Although specific approaches have been introduced to solve this problem, they can take between 3 and 10 minutes per game of runtime, may skip certain events, and potentially shuffle the order of events (Kim et al., 2025; Roy et al., 2024). `DataBallPy` allows for a state of the art synchronisation algorithm that ensures the synchronisation of all events in the right order within a few seconds (Oonk, Grob, et al., 2025) in just one line of code. Oonk, Grob, et al. (2025) showed that the expected goals model decreased in Brier loss from 0.096 to 0.082 (lower is better) when using the synchronisation in `DataBallPy` compared to a naive timestamp synchronisation. Similarly, the feature importance of features that relied on combined tracking and event data information was close to 0 in the timestamp synchronisation model, which was not the case for the `DataBallPy` synchronisation model (Oonk, Grob, et al., 2025).

Next to the practical value of `DataBallPy`, as it provides low-code access to scientific features (see the Features section below), `DataBallPy` also serves as an educational tool. Often, open-source Python packages provide information on how to get working code, but not on how the code works. `DataBallPy` explicitly goes a step further by elaborately explaining step by step how scientific papers are transformed into code, often referring to specific mathematical formulas as presented in the paper. These explanations are crucial since it (1) allows researchers and practitioners to better understand the strengths and weaknesses of features, and (2) teaches users how to transform scientific papers into modular, Pythonic code. Both these characteristics provide users of `DataBallPy` a better understanding of their own analysis.

Features

The features and functionalities in `DataBallPy` can be categorised into five categories: parsing data, preprocessing, synchronisation, performance indicators, and visualisation.

Parsing Data

The core goal of parsing data in `DataBallPy` is obtaining a `Game` object. `DataBallPy` allows for parsing data from different commercial data providers such as Tracab, Metrica, Inmotio, Opta, Instat, SciSports, Sportec, and Statsbomb internally using the `get_game` function. The `Game` object contains the event and tracking data internally as Pandas dataframes, making them intuitive to work with (team, 2020). Alternatively, one can use Kloppy to parse data from differnt providers and use the `get_game_from_kloppy` function to transform the Kloppy event and tracking datasets into a `Game` object. Last, `DataBallPy` has included a function to load openly available data directly in a `Game` object using `get_open_game`, which allows users who do not have access to data to still work with soccer data in `DataBallPy` (Bassek et al., 2025). Since the combination of parsing and (pre)processing a single game of data can take anywhere between 30 seconds and a few minutes on a standard device (which is similar to other packages), `DataBallPy` also allows one to efficiently save the preprocessed `Game` object as parquet and JSON files. This has two main benefits. First, using the `get_saved_game` function, you can now obtain a preprocessed game object in milliseconds instead of minutes, and second, raw tracking data files can be up to 400 MB per game, while the saved `DataBallPy Game` objects that include both event and tracking data are generally between 20 and 100 MB of memory.

Preprocessing

Tracking data is often captured via video footage using computer vision. Depending on the quality and number of cameras, some noise is present in both the athlete and ball positions (Linke et al., 2020). `DataBallPy` allows for filtering of the ball and positional data as well as differentiation of positions to compute velocity and acceleration. Furthermore, the tracking data allows for computation of individual athlete possession (Vidal-Codina et al., 2022), and together with the event data, team-level possession can be estimated.

Synchronisation

`DataBallPy` uses a soccer-specific implementation of the Needleman-Wunsch algorithm to synchronise the event and tracking data, which is more elaborately described in (Oonk, Grob, et al., 2025). The game can be synchronised using the following code

```
>>> from databallpy import get_open_game
>>> game = get_open_game()
>>> game.synchronise_tracking_and_event_data()
```

Performance Indicators

`DataBallPy` has an elaborate list of scientific features included in the package. All features can be computed in a few lines of code after obtaining a `Game` object. Next, the functionality, the documentation covers an elaborate explanation of how the code works that computes the features, which enables a clear reporting and reproduction of results. Using `DataBallPy`, the following features can be computed:

- Covered Distance (in specific velocity and acceleration zones) (Jerome et al., 2024)
- Pressure (Andrienko et al., 2017; Herold et al., 2022)
- Individual player possession (Vidal-Codina et al., 2022)
- Expected Goals (Anzer & Bauer, 2021)

- Expected Threat (Singh, 2019)
- Voronoi Space Occupation (Rein et al., 2017)
- Pitch Control (Fernandez & Bornn, 2018)
- Dangerous Accessible Space (Bischofberger & Baca, 2025)

Visualisation

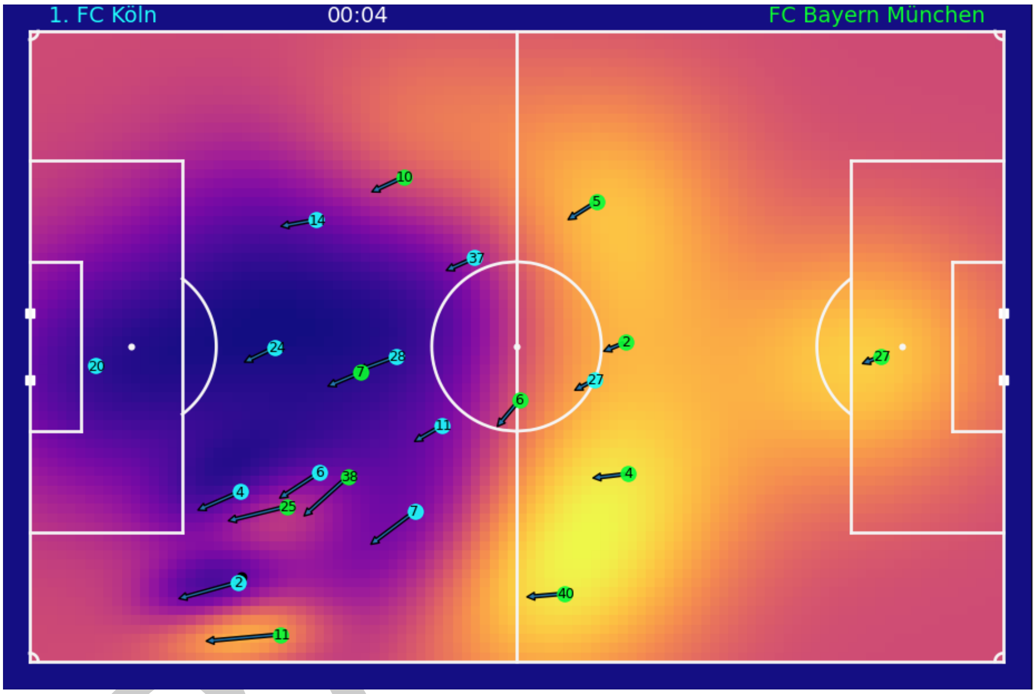

Figure 1: Example plot of soccer tracking data with pitch control heatmap as introduced in Fernandez & Bornn (2018)

DataBallPy includes elaborate functionality to visualise the data in the `Game` object. Event locations can be visualised on a pitch using the `plot_events()` function, which allows for coloring of events by outcome, team, or event type during specific periods in the game. Similarly, the locations and velocities of all players can be plotted using `plot_tracking_data()` function. If the event and tracking data are synchronised, one can also show information about the event in the same plot. Other features like pitch control heatmaps, player possession, and any custom feature can also be visualised simultaneously with the event and tracking data (Figure 1). Last, the tracking data (with heatmaps and custom features) can be transformed into a video (mp4) to show the true spatiotemporal progression over time.

```
import matplotlib.pyplot as plt

from databallpy import get_open_game
from databallpy.visualize import plot_tracking_data

game = get_open_game()
game.tracking_data.add_velocity(game.get_column_ids() + ["ball"])

pitch_control = game.tracking_data.get_pitch_control(
    game.pitch_dimensions,
    start_idx=100,
    end_idx = 101
```

```
)


fig, ax = plot_tracking_data(
    game,
    idx=100,
    add_velocities=True,
    heatmap_overlay=pitch_control[0],
    overlay_cmap="plasma",
    team_colors=["#00FFFF", "#00FF00"]
)
plt.show()
```

Research impact statement

DataBallPy has shown to be increasingly used by coders, practitioners, and researchers. The packages has been dowloaded over 47.000 times on PyPI, averaging more than 250 downloads per week. The project has over 60 GitHub stars. Issues and PR's are being opened by users outside of the network of the original owners and maintainers. On top of that, DataBallPy has been mentioned in numerous published scientific papers (Anzer et al., 2025; Oonk, Buurke, et al., 2025; Robertson et al., 2023; Zhang et al., 2025). Moreover, the largest currently open-sourced dataset of tracking and event data showcased how DataBallPy can be used to synchronise the two sources (Bassek et al., 2025). Last, authors that introduce new metrics propose to open a PR with their metric so it is easily available for the scientific community (Bischofberger & Baca, 2025). Together this shows that DataBallPy has a wide range of users and the package is growing outside of the reach of the original owners and maintainers.

AI usage disclosure

No generative AI tools were used in the writing of this manuscript and the development of the core functionalities and architecture of DataBallPy. With the exception of unittests, there is no explicit restriction on the usage of generative AI in the further development of DataBallPy (e.g. optimizing code, docstrings, reviewing, writing documentation, etc.). All code and documentation is checked and verified by human maintainers before merging into the code base.

References

Andrienko, G., Andrienko, N., Budziak, G., Dykes, J., Fuchs, G., Landesberger, T. von, & Weber, H. (2017). Visual analysis of pressure in football. *Data Mining and Knowledge Discovery*, *31*, 1793–1839. https://doi.org/10.1007/s10618-017-0513-2

Anzer, G., Arnsmeyer, K., Bauer, P., Bekkers, J., Brefeld, U., Davis, J., Evans, N., Kempe, M., Robertson, S. J., Smith, J. W., & Haaren, J. V. (2025). *Common data format (CDF): A standardized format for match-data in football (soccer)*. https://arxiv.org/abs/2505.15820v4

Anzer, G., & Bauer, P. (2021). A goal scoring probability model for shots based on synchronized positional and event data in football (soccer). *Frontiers in Sports and Active Living*, *3*, 624475. https://doi.org/https://doi.org/10.3389/fspor.2021.624475

Arem, K. van, Goes-Smit, F., & Söhl, J. (2025). Forecasting the future development in quality and value of professional football players. *Applied Sciences*, *15*(16), 8916. https://doi.org/10.3390/app15168916

Bassek, M., Rein, R., Weber, H., & Memmert, D. (2025). An integrated dataset of spatiotemporal and event data in elite soccer. *Scientific Data*, *12*, 195. https://doi.org/10.1038/S41597-025-04505-Y

Bauer, P., Anzer, G., & Shaw, L. (2023). Putting team formations in association football into context. *Journal of Sports Analytics*, *9*, 39–59. https://doi.org/10.3233/JSA-220620

Bischofberger, J., & Baca, A. (2025). *Dangerous accessible space: A unified model of space and value in team sports*. https://doi.org/10.21203/RS.3.RS-6932689/V1

Fernandez, J., & Bornn, L. (2018). Wide open spaces: A statistical technique for measuring space creation in professional soccer. *Sloan Sports Analytics Conference*. https://www.researchgate.net/publication/324942294_Wide_Open_Spaces_A_statistical_technique_for_measuring_space_creation_in_professional_soccer

Fernández, J., Bornn, L., & Cervone, D. (2021). A framework for the fine-grained evaluation of the instantaneous expected value of soccer possessions. *Machine Learning*, *110*, 1389–1427. https://doi.org/10.1007/S10994-021-05989-6

Forcher, L., Forcher, L., Altmann, S., Jekauc, D., & Kempe, M. (2022). The keys of pressing to gain the ball–characteristics of defensive pressure in elite soccer using tracking data. *Science and Medicine in Football*,. https://doi.org/10.1080/24733938.2022.2158213

Goes, F., Brink, M., Elferink-Gemser, M., Kempe, M., & Lemmink, K. A. P. M. (2020). The tactics of successful attacks in professional association football: Large-scale spatiotemporal analysis of dynamic subgroups using position tracking data. *Journal of Sports Sciences*, *39*, 523–532. https://doi.org/10.1080/02640414.2020.1834689

Goes, F., Meerhoff, L. A., Bueno, M. J. O., Rodrigues, D. M., Moura, F. A., Brink, M. S., Elferink-Gemser, M. T., Knobbe, A. J., Cunha, S. A., Torres, R. S., & Lemmink, K. A. P. M. (2020). Unlocking the potential of big data to support tactical performance analysis in professional soccer: A systematic review. *European Journal of Sport Science*, *21*, 481–496. https://doi.org/10.1080/17461391.2020.1747552

Goes, F., Schwarz, E., Elferink-Gemser, M., Lemmink, K., & Brink, M. (2021). A risk-reward assessment of passing decisions: Comparison between positional roles using tracking data from professional men's soccer. *Science and Medicine in Football*, *6*, 372–380. https://doi.org/10.1080/24733938.2021.1944660

Hader, K., Rumpf, M. C., Hertzog, M., Kilduff, L. P., Girard, O., & Silva, J. R. (2019). Monitoring the athlete match response: Can external load variables predict post-match acute and residual fatigue in soccer? A systematic review with meta-analysis. *Sports Medicine - Open*, *5*, 48–48. https://doi.org/10.1186/S40798-019-0219-7/FIGURES/2

Herold, M., Hecksteden, A., Radke, D., Goes, F., Nopp, S., Meyer, T., & Kempe, M. (2022). Off-ball behavior in association football: A data-driven model to measure changes in individual defensive pressure. *Journal of Sports Sciences*, *40*, 1412–1425. https://doi.org/10.1080/02640414.2022.2081405

Jerome, B. W. C., Stoeckl, M., Mackriell, B., Dawson, C. W., Fong, D. T. P., & Folland, J. P. (2024). Contextualised physical metrics: The physical demands vary with phase of play during elite soccer match play. *European Journal of Sport Science*, *24*, 1627–1638. https://doi.org/10.1002/EJSC.12209

Kim, H., Choi, H., Seo, S., Boomstra, T., Yoon, J., & Park, C. (2025). *ELASTIC: Event-tracking data synchronization in soccer without annotated event locations*. https://arxiv.org/abs/2508.09238

Link, D., Lang, S., & Seidenschwarz, P. (2016). Real time quantification of dangerousity in football using spatiotemporal tracking data. *PLOS ONE*, *11*, e0168768. https://doi.org/10.1371/JOURNAL.PONE.0168768

Linke, D., Link, D., & Lames, M. (2020). Football-specific validity of TRACAB's optical video tracking systems. *PLOS ONE*, *15*, e0230179. https://doi.org/10.1371/JOURNAL.PONE.0230179

Oonk, G. A., Buurke, T. J. W., Lemmink, K. A. P. M., & Kempe, M. (2025). The interaction between attacker and environment predicts successfulness in one-on-one dribbles in male elite football. *Journal of Sports Sciences*. https://doi.org/10.1080/02640414.2025.2555117

Oonk, G. A., Grob, D., & Kempe, M. (2025). The right way to synchronize tracking and event data: Using domain knowledge to optimize algorithms. In D. Goossens (Ed.), *MathSports conference* (pp. 136–143).

Raabe, D., Biermann, H., Bassek, M., Wohlan, M., Komitova, R., Rein, R., Groot, T. K., & Memmert, D. (2022). Floodlight - a high-level, data-driven sports analytics framework. *Journal of Open Source Software*, *7*(76), 4588. https://doi.org/10.21105/joss.04588

Rein, R., Raabe, D., & Memmert, D. (2017). "Which pass is better?" Novel approaches to assess passing effectiveness in elite soccer. *Human Movement Science*, *55*, 172–181. https://doi.org/10.1016/j.humov.2017.07.010

Robertson, S., Duthie, G. M., Ball, K., Spencer, B., Serpiello, F. R., Haycraft, J., Evans, N., Billingham, J., & Aughey, R. J. (2023). Challenges and considerations in determining the quality of electronic performance & tracking systems for team sports. *Frontiers in Sports and Active Living*, *5*, 1266522. https://doi.org/10.3389/FSPOR.2023.1266522/BIBTEX

Roy, M. V., Cascioli, L., & Davis, J. (2024). ETSY: A rule-based approach to event and tracking data SYnchronization. *Communications in Computer and Information Science*, *2035 CCIS*, 11–23. https://doi.org/10.1007/978-3-031-53833-9_2

Singh, K. (2019). *Introducing expected threat (xT)*. https://karun.in/blog/expected-threat.html

Sotudeh, H. (2025). The principles of tactical formation identification in association football (soccer)—a survey. *Frontiers in Sports and Active Living*, *6*, 1512386.

Spearman, W., Basye, A. T., Basye, A., Dick, G., Hotovy, R., & Pop, P. (2017). Physics-based modeling of pass probabilities in soccer. *Sports Analytics Conference*. https://www.researchgate.net/profile/William-Spearman/publication/315166647_Physics-Based_Modeling_of_Pass_Probabilities_in_Soccer/links/58cbfca2aca272335513b33c/Physics-Based-Modeling-of-Pass-Probabilities-in-Soccer.pdf

team, T. pandas development. (2020). *Pandas-dev/pandas: pandas* (latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Vidal-Codina, F., Evans, N., Fakir, B. E., & Billingham, J. (2022). Automatic event detection in football using tracking data. *Sports Engineering*, *25*. https://doi.org/10.1007/s12283-022-00381-6

Zhang, G., Kempe, M., McRobert, A., Folgado, H., & Olthof, S. B. H. (2025). Navigating team tactical analysis in football: An analytical pipeline leveraging player tracking technology. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*. https://doi.org/10.1177/17543371251392456/SUPPL_FILE/SJ-DOCX-1-PIP-10.1177_17543371251392456.DOCX