The Journal of Open Source Software

floodlight - A high-level, data-driven sports analytics framework

Dominik Raabe[1¶], **Henrik Biermann**[1], **Manuel Bassek**[1], **Martin Wohlan**[1], **Rumena Komitova**[1], **Robert Rein**[1], **Tobias Kuppens Groot**[2], **and Daniel Memmert**[1]

1 Institute of Exercise Training and Sport Informatics, German Sport University Cologne, Germany **2** Independent Researcher, Netherlands ¶ Corresponding author

DOI: 10.21105/joss.04588

Software

- Review
- Repository
- Archive

Editor: Chris Vernon

Reviewers:

- @gagolews
- @kanishkan91

Submitted: 08 June 2022
Published: 23 August 2022

Summary

The increase of available data has had a positive impact on the entire sports domain and especially sport science (Morgulev et al., 2018). Two major data sources of relevance in this domain are spatiotemporal tracking data of athlete positions as well as manually annotated match event data (Memmert & Raabe, 2018; Stein et al., 2017). These two data types are regularly collected by professional sport organizations in different team invasion games such as football, basketball, or handball (Memmert, 2021). These data sources open up a whole range of new analysis possibilities across multiple (sub)disciplines in the field, including match and performance analysis, exercise physiology, training science, or collective movement behavior analysis. As an example, player tracking data has been used extensively to analyze physical (Castellano et al., 2014) as well as tactical (Rein & Memmert, 2016) performance in football.

The *floodlight* Python package provides a framework to support and automate team sport data analysis. *floodlight* is constructed to process spatiotemporal tracking data, event data, and other game meta-information to support scientific performance analyses. *floodlight* was designed to provide a general yet flexible approach to performance analysis, while simultaneously providing a user-friendly high-level interface for users with basic programming skills. The package includes routines for most aspects of the data analysis process, including dedicated data classes, file parsing functionality, public dataset APIs, pre-processing routines, common data models and several standard analysis algorithms previously used in the literature, as well as basic visualization functionality.

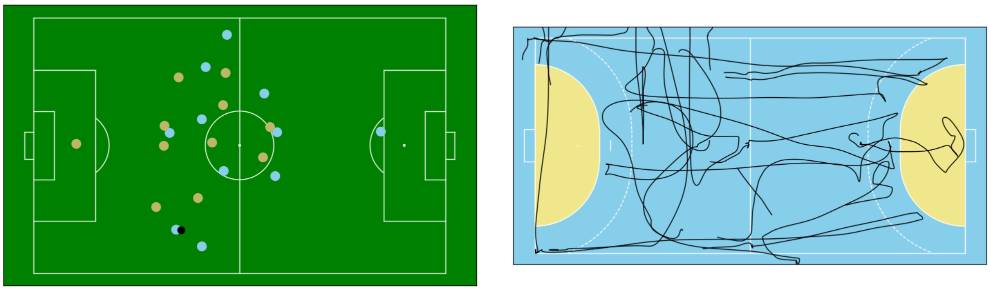

Figure 1: Positions of football players (left) and trajectories of handball players (right) from real-world match data as visualized with *floodlight*.

Features

Data-level objects: XY (tracking data), Events (event data), Code (meta information), Pitch (pitch layout), PlayerProperty (player information per frame), TeamProperty (team information per frame), DyadicProperty (player interaction information per frame).

Data parser: For provider raw data files from ChyronHego (tracking data, codes), DFL (tracking data, codes), Kinexon (tracking data), Opta (event data), Second Spectrum (tracking data), StatsPerform (tracking data, event data) StatsBomb (event data).

Datasets: EIGD-H, StatsBomb OpenData.

Processing: Spatial transforms (tracking and event data), Butterworth and Savitzky-Golay lowpass filter (tracking data), data slicing (all temporal objects), selection and sequencing (event data, codes).

Visualization: Pitches (football and handball), player positions, player trajectories.

Data models: Distances, distances covered, velocities, accelerations, centroids, centroid distances, stretch index, metabolic power, equivalent distances, approximate entropy.

Documentation: Module reference, extended contributing guide, team sports data analysis compendium, tutorials (getting started, analyzing data, preparing match sheets).

Central to the package is a set of generalized, provider- and sports-independent core data structures based on *NumPy* (Harris et al., 2020) and *pandas* (McKinney, 2010). Each of these data structures are dedicated to one specific type of sports data, including spatiotemporal tracking data, event data, game codes (meta information such as ball possession information), pitch information regarding the embedding of data and playing surfaces into Cartesian coordinate systems, as well as team and player properties (such as frame-wise velocity or acceleration values). The data structures are designed with a focus on scientific computing, i.e., optimized for accessible and intuitive data manipulations as well as sensitive to performance by utilizing *NumPy*'s view-, vectorization- and indexing techniques.

The core data classes allow internal storage and processing of sports data whilst decoupling from any format-specific requirements. Consequently, *floodlight* is built around these objects, comprising several elementary modules of the data processing pipeline. For data loading, the package provides parsing submodules with functions that dissect and map data from specific provider formats to core data structures (including providers such as Kinexon, Tracab, Stats Perform, StatsBomb, Second Spectrum, Opta, or DFL), which eliminates problems caused by the many, strongly varying data formats in use. Data loaders and mappers for available public datasets such as the EIGD-H dataset (Biermann et al., 2021) are additionally included. In terms of data processing, the package provides dedicated manipulation functionality such as spatial transformations helpful for spatial data synchronization or signal filters based on *SciPy* (Virtanen et al., 2020). For data inspection, basic visualization functionality based on the *Matplotlib* package (Hunter, 2007) is included (see Figure 1).

The actual data analysis part is realized by a submodule providing several data models. These models provide a toolbox of domain-specific data analysis procedures from different subdomains such as exercise physiology, e.g., the metabolic power model (di Prampero & Osgnach, 2018), dynamical system approaches, e.g., approximate entropy (Pincus, 1991), or collective tactical behavior, e.g., centroid-based measures (Bourbousson et al., 2010; Sampaio & Maçãs, 2012). All models follow the same syntax inspired by the *scikit-learn* package (Buitinck et al., 2013), where upon instantiation, a central fitting method is called with core data structures. Subsequently, required computations can be queried with additional class methods. This allows a consistent syntax and collection of similar measures into cohesive data models while limiting the repetition of basic calculations and allowing simple future extensions.

The following code sample illustrates how *floodlight* reduces a typical performance analysis pipeline to just a few lines of code. In the example, one sample of data is queried from the public EIGD-H dataset, filtered, and the cumulative metabolic work of the home team is calculated for the entire segment of data:

```
from floodlight.io.datasets import EIGDDataset
from floodlight.transforms.filter import butterworth_lowpass
from floodlight.models.kinetics import MetabolicPowerModel

dataset = EIGDDataset()
home_team_data, away_team_data, ball_data = dataset.get()

home_team_data = butterworth_lowpass(home_team_data)

model = MetabolicPowerModel()
model.fit(home_team_data)
metabolic_power = model.cumulative_metabolic_power()
```

These lines of code create a new core object named `metabolic_power`, which contains an array of shape (T x N) storing (cumulated) metabolic power values (in joule per kilogram) for each of the T time points and N players. For example, we can print the total metabolic work of the seven active players:

```
>>> print(metabolic_power[-1, 0:7])

[1669.18781115 1536.22481121 1461.03243489 1488.61249785  773.09264071
 1645.01702421  746.94057676]
```

Statement of need

Despite the increase in volume, the technical requirements for team sport data analysis have constantly remained high. This can be partially attributed to the complexity and heterogeneity of the data itself (Memmert & Raabe, 2018; Stein et al., 2017), but also to multiple practical and theoretical challenges. These include the necessity of complex file parsing procedures for provider-specific data formats, low compatibility across data providers, or differing standards for spatial or temporal resolution of data, often requiring specialized pre-processing routines. Meeting these challenges typically requires massive and customized overhead programming in sports data analysis projects. At the same time, there hardly exist any general, proprietary or open source, software alternatives which can be used out of the box for scientific purposes. Existing software is either commercially driven (i.e., proprietary, limited to a specific data provider or focused on industrial applications), or task-specific (i.e., limited to a certain data source, data format, sport or subtask) which leaves the problem of adapting code to multiple different APIs within the analysis process.

These current constraints resulted in a situation where a typical analysis workflow requires the (re)implementation of each processing pipeline module in its entirety with respect to the specific project's needs. For sport scientists who typically lack programming skills (which are usually not part of their formal training) this can become an insurmountable hurdle. As a consequence, advanced team sports data analyses remain inaccessible for large parts of the sport scientific community which poses a significant hindrance for future progress. Accordingly, the *floodlight* package was designed to specifically address this problem and significantly ease advanced analyses of sports data. *floodlight* automates standard data processing routines and provides a high-level interface accessible to users with just basic programming skills. The *floodlight* documentation contains several tutorials as well as an extensive compendium discussing the technical aspects of team sports data analysis to ensure easy access and understanding of the routines and their design choices. The tutorials increase the beginner-friendliness of *floodlight* and allow its usage in educational settings, e.g., for team sport data analytics courses.

Another hurdle faced by sports scientists relates to the current lack of collaboration and code sharing practices within the field. At present, sharing proposed data models or algorithms for analyses is the exception rather than the rule. In parts, the lack of sharing often stems from the proprietary nature of the raw data, but is further exacerbated by lack of data format gold standards. More generally, disciplines that employ team sports data analysis have reported a culture that contains very little replications and works incorporating previous findings (Herold et al., 2019), low applicability of research by practitioners (Bishop, 2008; Herold et al., 2019; Mackenzie & Cushion, 2013) and limited interdisciplinary approaches between computer and sport scientists (Goes et al., 2021; Rein & Memmert, 2016). A major milestone in the process of meeting these challenges is to find feasible ways of sharing data and algorithms (Rein & Memmert, 2016). The *floodlight* package can be seen as a first step in this direction with a toolbox-approach collecting common data manipulation and processing techniques.

floodlight will therefore be equally useful for sports scientists as well as computer scientists, working in academia or applied settings. The package will therefore serve to bring these users groups together and foster future interdisciplinary collaborations. Ideally, this will also promote further open source contributions that share advanced data processing algorithms in the domain and enable future work incorporating previous findings.

Acknowledgements

This project has received funding from the German Federal Ministry of Education and Research (BMBF) to the last author under grant number 01IS20021A.

References

Biermann, H., Theiner, J., Bassek, M., Raabe, D., Memmert, D., & Ewerth, R. (2021). A Unified Taxonomy and Multimodal Dataset for Events in Invasion Games. *Proceedings of the 4th International Workshop on Multimedia Content Analysis in Sports*, 1–10. https://doi.org/10.1145/3475722.3482792

Bishop, D. (2008). An Applied Research Model for the Sport Sciences. *Sports Medicine*, *38*(3), 253–263. https://doi.org/10.2165/00007256-200838030-00005

Bourbousson, J., Sève, C., & McGarry, T. (2010). Space–time coordination dynamics in basketball: Part 2. The interaction between the two teams. *Journal of Sports Sciences*, *28*(3), 349–358. https://doi.org/10.1080/02640410903503640

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). *API design for machine learning software: Experiences from the scikit-learn project*. https://doi.org/10.48550/ARXIV.1309.0238

Castellano, J., Alvarez-Pastor, D., & Bradley, P. S. (2014). Evaluation of research using computerised tracking systems (amisco® and prozone®) to analyse physical performance in elite soccer: A systematic review. *Sports Medicine*, *44*(5), 701–712. https://doi.org/10.1007/s40279-014-0144-3

di Prampero, P., & Osgnach, C. (2018). Metabolic Power in Team Sports - Part 1: An Update. *International Journal of Sports Medicine*, *39*(08), 581–587. https://doi.org/10.1055/a-0592-7660

Goes, F. R., 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. (2021). Unlocking the potential of big data to support tactical performance analysis in professional soccer: A systematic review. *European Journal of Sport Science*, *21*(4), 481–496. https://doi.org/10.1080/17461391.2020.1747552

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Herold, M., Goes, F., Nopp, S., Bauer, P., Thompson, C., & Meyer, T. (2019). Machine learning in men's professional football: Current applications and future directions for improving attacking play. *International Journal of Sports Science and Coaching*, *14*(6), 798–817. https://doi.org/10.1177/1747954119879350

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Mackenzie, R., & Cushion, C. (2013). Performance analysis in football: A critical review and implications for future research. *Journal of Sports Sciences*, *31*(6), 639–676. https://doi.org/10.1080/02640414.2012.746720

McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Python in Science Conference*, 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

Memmert, D. (2021). *Match Analysis: How to Use Data in Professional Sport*. Routledge. https://doi.org/10.4324/9781003160953

Memmert, D., & Raabe, D. (2018). Data Analytics in Football: Positional Data Collection, Modelling and Analysis. In *Data Analytics in Football*. Routledge. https://doi.org/10.4324/9781351210164

Morgulev, E., Azar, O. H., & Lidor, R. (2018). Sports analytics and the big-data era. *International Journal of Data Science and Analytics*, *5*(4), 213–222. https://doi.org/10.1007/s41060-017-0093-7

Pincus, S. (1991). Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, *88*(6), 2297–2301. https://doi.org/10.1073/pnas.88.6.2297

Rein, R., & Memmert, D. (2016). Big data and tactical analysis in elite soccer: Future challenges and opportunities for sports science. *SpringerPlus*, *5*(1), 1410–1410. https://doi.org/10.1186/s40064-016-3108-2

Sampaio, J., & Maçãs, V. (2012). Measuring tactical behaviour in football. *International Journal of Sports Medicine*, *33*(5), 395–401. https://doi.org/10.1055/s-0031-1301320

Stein, M., Janetzko, H., Seebacher, D., Jäger, A., Nagel, M., Hölsch, J., Kosub, S., Schreck, T., Keim, D., & Grossniklaus, M. (2017). How to Make Sense of Team Sport Data: From Acquisition to Data Modeling and Research Aspects. *Data*, *2*(1), 2. https://doi.org/10.3390/data2010002

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2