The Journal of Open Source Software

# TimeSeriesClustering: An extensible framework in Julia

**Holger Teichgraeber**[1], **Lucas Elias Kuepper**[1], **and Adam R. Brandt**[1]

**1** Department of Energy Resources Engineering, Stanford University

**DOI:** 10.21105/joss.01573

**Software**

- Review
- Repository
- Archive

**Submitted:** 10 July 2019
**Published:** 08 September 2019



## Summary

`TimeSeriesClustering` is a Julia implementation of unsupervised learning methods for time series datasets. It provides functionality for clustering and aggregating, detecting motifs, and quantifying similarity between time series datasets. The software provides a type system for temporal data, and provides an implementation of the most commonly used clustering methods and extreme value selection methods for temporal data. `TimeSeriesClustering` provides simple integration of multi-dimensional time-series data (e.g., multiple attributes such as wind availability, solar availability, and electricity demand) in a single aggregation process. The software is applicable to general time series datasets and lends itself well to a multitude of application areas within the field of time series data mining. `TimeSeriesClustering` was originally developed to perform time series aggregation for energy systems optimization problems. Because of the software's origin, many of the examples in this work stem from the field of energy systems optimization.

### General package features

The unique design of `TimeSeriesClustering` allows for scientific comparison of the performance of different time-series aggregation methods, both in terms of the statistical error measure and in terms of its impact on the application outcome. The clustering methods that are implemented in `TimeSeriesClustering` follow the framework presented by Teichgraeber & Brandt (2019), and the extreme value selection methods follow the framework presented by Lindenmeyer et al. (2020). Using these frameworks allows `TimeSeriesClustering` to be generally extensible to new aggregation methods in the future.

The following are the key features that `TimeSeriesClustering` provides. Implementation details can be found in the software's documentation.

- *The type system*: The data type (called struct in Julia) `ClustData` stores all time-series data in a common format. Besides the data itself, it automatically processes and stores information that is relevant for later use in the application for which the time-series data will be used. The data type `ClustResult` additionally stores information relevant for evaluating clustering performance. These data types make `TimeSeriesClustering` easy to integrate with any analysis that relies on iterative evaluation of the clustering and aggregation methods.
- *The aggregation methods*: The most commonly used clustering methods and extreme value selection methods are implemented with a common interface, allowing for simple comparison of these methods on a given data set and optimization problem.
- *The generalized import of time series in csv format*: Time series can be loaded through csv files in a pre-defined format. From this, variable names, which we call attributes, and node names are automatically loaded and stored. The original time series can be sliced into periods of user-defined length. This information can later be used in the definition of the sets of the optimization problem.

- *Multiple attributes and nodes*: Multiple time series, one for each attribute (and node, if the data has a spatial component), are automatically combined and aggregated simultaneously.

## Package features useful for energy systems optimization

`TimeSeriesClustering` was originally developed for time-series input data to energy systems optimization problems. In this section, we describe some of its features with respect to their use in energy systems optimization.

In energy systems optimization, the choice of temporal modeling, especially of time-series aggregation methods, can have significant impact on overall optimization outcome, which in the end is used to make policy and business decisions. It is thus important to not view time-series aggregation and optimization model formulation as two seperate, consecutive steps, but to integrate time-series aggregation into the overall process of building an optimization model in an iterative manner. Because the most commonly used clustering methods and extreme value selection methods are implemented with a common interface, `TimeSeriesClustering` allows for this iterative integration in a simple way.

The type system for temporal data provided by `TimeSeriesClustering` allows for easy integration with the formulation of optimization problems. The information stored in the datatype `ClustData`, such as the number of periods, the number of time steps per period, and the chronology of the periods, can be used to formulate the sets of an optimization problem.

`TimeSeriesClustering` provides two sample optimization problems to illustrate the integration of time-series aggregation and optimization problem formulation through our type system. However, it is generally thought to be independent of the application at hand, and others are encouraged to use the package as a base for their own optimization problem formulation. The Julia package `CapacityExpansion` provides a detailed generation and transmission capacity expansion model built upon `TimeSeriesClustering`, and illustrates its capabilities in conjunction with a complex optimization problem formulation.

## `TimeSeriesClustering` within the broader ecosystem

`TimeSeriesClustering` is the first package to provide broadly applicable unsupervised learning methods specifically for time series in Julia (Bezanson, Edelman, Karpinski, & Shah, 2017). There are several other related packages that provide useful tools for these tasks, both in Julia and in the general open-source community, and we describe them in order to provide guidance on the broader tools available for these kinds of modeling problems.

The `Clustering` package in Julia provides a broad range of clustering methods and and allows computation of clustering validation measures. `TimeSeriesClustering` provides a simplified workflow for clustering time series, and works on top of the `Clustering` package by making use of a subset of the clustering methods implemented in the `Clustering` package. `TimeSeriesClustering` has several features that add to the functionality, such as automatically clustering multiple attributes simultaneously and providing multiple initializations for partitional clustering algorithms.

The `TSML` package in Julia provides processing and machine learning methods for time-series data. Its focus is on time-series data with date and time stamps, and it provides a broad range of processing tools. It integrates with other machine learning libraries within the broader Julia ecoysystem.

The `TimeSeries` package in Julia provides a way to store data with time stamps, and perform table opertions and plotting based on time stamps. The `TimeSeries` package may be useful

for pre-processing or post-processing data in conjunction with TimeSeriesClustering. The main difference is in the way data is stored: In the TimeSeries package, data is stored based on time stamps. In TimeSeriesClustering, we store data based on index and time step length, which is relevant to clustering and its applications.

In python, clustering and time-series analysis tasks can be performed using packages such as scikit-learn (Pedregosa et al., 2011) and PyClustering (Novikov, 2019). The package tslearn provides clustering methods specifically for time series, both the conventional k-means method and shape-based methods such as k-shape and dynamic time warping barycenter averaging. The STUMPY package (Law, 2019) calculates something called the matrix profile, which can be used for many data mining tasks.

In R, time series clustering can be performed using the tsclust package (Montero & Vilar, 2014), and the dtw package (Giorgino, 2009) provides functionality for dynamic time warping, i.e. when the shape of the time series matters for clustering.

With specific focus on energy systems optimization, time-series aggregation has been included in two open-source packages to date, both in written in Python. TSAM (Kotzur & Kaldemeyer, 2017) provides an implementation of several time-series aggregation methods in Python. Calliope (Pfenninger & Pickering, 2018) is a capacity expansion modeling software in Python that includes time-series aggregation for the use case of generation and transmission capacity expansion modeling. TimeSeriesClustering is the first package to provide time-series aggregation in Julia (Bezanson et al., 2017). For energy systems optimization, this is advantageous because it can be used in conjunction with the JuMP package (Dunning, Huchette, & Lubin, 2017) in Julia, which provides an excellent modeling language for optimization problems. Furthermore, TimeSeriesClustering includes both clustering and extreme value selection, and integrates them into the same output type. This is important in order to retain the characteristics of the time-series that are relevant to many optimization problems.

## Application areas

TimeSeriesClustering is broadly applicable to many fields where time series analysis occurs. Time-series clustering and aggregation methods alone have applications in the fields of aviation, astronomy, biology, climate, energy, environment, finance, medicine, psychology, robotics, speech recognition, and user analysis (Warren Liao, 2005, Aghabozorgi, Seyed Shirkhorshidi, & Ying Wah (2015)). These methods can be used for time-series representation and indexing, which helps reduce the dimension (i.e., the number of data points) of the original data (Fu, 2011).

Many tasks in time series data mining also fall into the application area of our software (Fu, 2011, Hebert, Anderson, Olinsky, & Hardin (2014)). Here, our software can be used to measure similarity between time-series datasets (Serrà & Arcos, 2014). Closely related is the task of finding time-series motifs (Lin, Keogh, Lonardi, & Patel, 2002, Yankov, Keogh, Medina, Chiu, & Zordan (2007), Mueen (2014)). Time-series motifs are pairs of individual time series that are very similar to each other. This task occurs in many disciplines, for example in finding repeated animal behavior (Mueen, Keogh, Zhu, Cash, & Westover, 2013), finding regulatory elements in DNA (Das & Dai, 2007), and finding patterns in EEG signals (Castro & Azevedo, 2010). Another application area of our software is segmentation and clustering of audio datasets (Siegler, Jain, Raj, & Stern, 1997, Lefèvre & Vincent (2011), Kamper, Livescu, & Goldwater (2017)).

In the remainder of this section, we provide an overview of how time-series aggregation applies to the input data of optimization problems.

Generally, optimization is concerned with the maximization or minimization of a certain objective subject to a number of constraints. The range of optimization problems TimeSer

iesClustering is applicable to is broad. They generally fall into the class of design and operations problems, also called planning problems or two-stage optimization problems. In these problems, decisions on two time horizons have to be made: long-term design decisions, as to what equipment to buy, and short-term operating decisions, as to when to operate that equipment. Because the two time horizons are intertwined, operating decisions impact the system design, and vice versa. Operating decisions are of a temporal nature, and the amount of temporal input data for these optimization problems often makes them computationally intractable. Usually, time series of length $N$ (e.g., hourly electricity demand for one year, where $N = 8760$) are split into $\hat{K}$ periods of length $T = \frac{N}{\hat{K}}$ (e.g., $\hat{K} = 365$ daily periods, with $T = 24$), and each of the $\hat{K}$ periods is treated independently in the operations stage of the optimization problem. Using time-series aggregation methods, we can represent the data with $K < \hat{K}$ periods, which results in reduced computational complexity and improved modeling performance.

Many of the design and operations optimization problems to which time-series aggregation has been applied are in the general domain of energy systems optimization. These problems include generation and transmission capacity expansion problems (Nahmmacher, Schmid, Hirth, & Knopf, 2016; Pfenninger, 2017), local energy supply system design problems (Bahl, Kümpel, Seele, Lampe, & Bardow, 2017; Kotzur, Markewitz, Robinius, & Stolten, 2018), and individual technology design problems (Brodrick, Brandt, & Durlofsky, 2017; Teichgraeber, Brodrick, & Brandt, 2017). Time series of interest in these problems include energy demands (electricity, heating, cooling), electricity prices, wind and solar availability factors, and temperatures.

Many other planning problems in operations research that involve time-varying operations have similar characteristics that make them suitable for time-series aggregation. Some examples are aggregate and detailed production scheduling, job shop design and scheduling, distribution system (warehouse) design and control (Dempster et al., 1981), and electric vehicle charging station sizing (Jia, Hu, Song, & Luo, 2012). Time series of interest in these problems include product demands, electricity prices, and electricity demands. A related class of problems to which `TimeSeriesClustering` can be useful is scenario reduction for stochastic programming (Karuppiah, Martín, & Grossmann, 2010). Two-stage stochastic programs have similar characteristics to the previously described two-stage problems, and are often computationally intractable due to a large number of scenarios. `TimeSeriesClustering` can be used to reduce a large number of scenarios $\hat{K}$ into a computationally tractable number of scenarios $K < \hat{K}$. Furthermore, `TimeSeriesClustering` could be used in operational contexts such as developing operational strategies for typical days, or aggregating repetitive operating conditions for use in model predictive control. Because it keeps track of the chronology of the periods, it can also be used to calculate transition probabilities between clustered periods for Markov chain modeling.

`TimeSeriesClustering` has been used in several research projects to date. It has been used to compare both conventionally-used clustering methods and shape-based clustering methods and their characteristics (Teichgraeber & Brandt, 2019), and also to compare extreme value selection methods (Lindenmeyer et al., 2020). It has also been used to analyze temporal modeling detail in energy systems modeling with high renewable energy penetration (Kuepper, 2019). `TimeSeriesClustering` also serves as input to `CapacityExpansion`, a scalable capacity expansion model in Julia. Furthermore, `TimeSeriesClustering` has been used as an educational tool. It is frequently used for class projects in the Stanford University course "Optimization of Energy Systems", and has also served as a basis for the capacity expansion studies evaluated in homeworks for the Stanford University course "Advanced Methods in Modeling for Climate and Energy Policy".

## References


Aghabozorgi, S., Seyed Shirkhorshidi, A., & Ying Wah, T. (2015). Time-series clustering - A decade review. *Information Systems*, *53*, 16–38. doi:10.1016/j.is.2015.04.007

Bahl, B., Kümpel, A., Seele, H., Lampe, M., & Bardow, A. (2017). Time-series aggregation for synthesis problems by bounding error in the objective function. *Energy*, *135*, 900–912. doi:10.1016/j.energy.2017.06.082

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, *59*(1), 65–98. doi:10.1137/141000671

Brodrick, P. G., Brandt, A. R., & Durlofsky, L. J. (2017). Operational optimization of an integrated solar combined cycle under practical time-dependent constraints. *Energy*, *141*, 1569–1584. doi:10.1016/j.energy.2017.11.059

Castro, N., & Azevedo, P. (2010). Multiresolution motif discovery in time series. *Proceedings of the 10th SIAM International Conference on Data Mining, SDM 2010*, 665–676. doi:https://doi.org/10.1137/1.9781611972801.73

Das, M. K., & Dai, H. K. (2007). A survey of DNA motif finding algorithms. *BMC Bioinformatics*, *8*, 1–13. doi:10.1186/1471-2105-8-S7-S21

Dempster, M. A. H., Fisher, M. L., L., J., B.J., L., J.K., L., & A.H.G., R. K. (1981). Analytical Evaluation of Hierarchical planning systems. *Operations Research*, *29*(4), 707–716.

Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, *59*(2), 295–320. doi:10.1137/15M1020575

Fu, T. C. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, *24*(1), 164–181. doi:10.1016/j.engappai.2010.09.007

Giorgino, T. (2009). Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software*, *31*(7), 1–24. doi:10.18637/jss.v031.i07

Hebert, D., Anderson, B., Olinsky, A., & Hardin, J. M. (2014). Time Series Data Mining. *International Journal of Business Analytics*, *1*(4), 51–68. doi:10.4018/ijban.2014100104

Jia, L., Hu, Z., Song, Y., & Luo, Z. (2012). Optimal siting and sizing of electric vehicle charging stations. *2012 IEEE International Electric Vehicle Conference, IEVC 2012*, 1–6. doi:10.1109/IEVC.2012.6183283

Kamper, H., Livescu, K., & Goldwater, S. (2017). An embedded segmental k-means model for unsupervised segmentation and clustering of speech. *IEEE Workshop on Automatic Speech Recognition and Understanding*, 719–726. doi:https://doi.org/10.1109/asru.2017.8269008

Karuppiah, R., Martín, M., & Grossmann, I. E. (2010). A simple heuristic for reducing the number of scenarios in two-stage stochastic programming. *Computers and Chemical Engineering*, *34*(8), 1246–1255. doi:10.1016/j.compchemeng.2009.10.009

Kotzur, L., & Kaldemeyer, C. (2017). TSAM. doi:10.5281/zenodo.2547683

Kotzur, L., Markewitz, P., Robinius, M., & Stolten, D. (2018). Impact of different time series aggregation methods on optimal energy system design. *Renewable Energy*, *117*, 474–487. doi:10.1016/j.renene.2017.10.017

Kuepper, L. E. (2019). *Investigation of time-series aggregation methods for infrastructure optimization of low emissions energy systems* (MS thesis). Stanford University, RWTH Aachen University, Aachen.

Law, S. (2019). STUMPY: A Powerful and Scalable Python Library for Time Series Data Mining. *Journal of Open Source Software*, *4*(39), 1504. doi:10.21105/joss.01504

Lefèvre, S., & Vincent, N. (2011). A two level strategy for audio segmentation. *Digital Signal Processing: A Review Journal*, *21*(2), 270–277. doi:10.1016/j.dsp.2010.07.003

Lin, J., Keogh, E., Lonardi, S., & Patel, P. (2002). Finding motifs in time series. *Proc. of the 2nd Workshop on Temporal Data Mining*, 53–68.

Lindenmeyer, C., Teichgraeber, H., Baumgaertner, N., Kotzur, L., Robinius, M., Bardow, A., & Brandt, A. R. (2020). Extreme events as part of representative periods for the optimization of residential energy supply systems. *Energy (Manuscript in preparation)*.

Montero, P., & Vilar, J. A. (2014). TSclust: An R package for time series clustering. *Journal of Statistical Software*, *62*(1), 1–43. doi:10.18637/jss.v062.i01

Mueen, A. (2014). Time series motif discovery: Dimensions and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(2), 152–159. doi:10.1002/widm.1119

Mueen, A., Keogh, E., Zhu, Q., Cash, S., & Westover, B. (2013). Exact Discovery of Time Series Motifs, 473–484. doi:10.1137/1.9781611972795.41

Nahmmacher, P., Schmid, E., Hirth, L., & Knopf, B. (2016). Carpe diem: A novel approach to select representative days for long-term power system modeling. *Energy*, *112*, 430–442. doi:10.1016/j.energy.2016.06.081

Novikov, A. (2019). PyClustering: Data Mining Library. *Journal of Open Source Software*, *4*(36), 1230. doi:10.21105/joss.01230

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pfenninger, S. (2017). Dealing with multiple decades of hourly wind and PV time series in energy models: A comparison of methods to reduce time resolution and the planning implications of inter-annual variability. *Applied Energy*, *197*, 1–13. doi:10.1016/j.apenergy.2017.03.051

Pfenninger, S., & Pickering, B. (2018). Calliope: a multi-scale energy systems modelling framework. *Journal of Open Source Software*, *3*(29), 825. doi:10.21105/joss.00825

Serrà, J., & Arcos, J. L. (2014). An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems*, *67*, 305–314. doi:10.1016/j.knosys.2014.04.035

Siegler, M. A., Jain, U., Raj, B., & Stern, R. M. (1997). Automatic Segmentation, Classification and Clustering of Broadcast News Audio. *Proc. DARPA Speech Recognition Workshop*, 97–99.

Teichgraeber, H., & Brandt, A. R. (2019). Clustering methods to find representative periods for the optimization of energy systems : an initial framework and comparison. *Applied Energy*, *239*, 1283–1293. doi:10.1016/j.apenergy.2019.02.012

Teichgraeber, H., Brodrick, P. G., & Brandt, A. R. (2017). Optimal design and operations of a flexible oxyfuel natural gas plant. *Energy*, *141*, 506–518. doi:10.1016/j.energy.2017.09.087

Warren Liao, T. (2005). Clustering of time series data—a survey. *Pattern Recognition*, *38*(11), 1857–1874. doi:10.1016/j.patcog.2005.01.025

Yankov, D., Keogh, E., Medina, J., Chiu, B., & Zordan, V. (2007). Detecting time series motifs under uniform scaling. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 844–853. doi:10.1145/1281192.1281282