Computers & Graphics (2023)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

LFPeers: Temporal Similarity Search and Result Exploration

Madhav Sachdeva[c,1,*], Jan Burmeister[a], Jörn Kohlhammer[a,b], Jürgen Bernard[c,d,2]

[a] *Fraunhofer IGD, Darmstadt, Germany*
[b] *Technische Universität Darmstadt, Darmstadt, Germany*
[c] *University of Zurich, Zurich, Switzerland*
[d] *Digital Society Initiative, Zurich, Switzerland*

ARTICLE INFO

Article history:
Received July 25, 2023

Keywords: Search and Exploration, Temporal Similarity, Visual Analytics, Nearest Neighbor Search, Relation Seeking,, Relation Explanation

ABSTRACT

In this paper, we introduce a general concept for the analysis of temporal and multivariate data and the system LFPeers that applies this concept to temporal similarity search and results exploration. The conceptual workflow divides the analysis in two phases: a search phase to find the most similar objects to a query object before a time point t_0 in the temporal data, and an exploration phase to analyze and contextualize this subset of objects after t_0. LFPeers enables users to search for peers through interactive similarity search and filtering, explore interesting behavior of this peer group, and learn from peers through the assessment of diverging behaviors. We present the conceptual workflow to learn from peers and the LFPeers system with novel interfaces for search and exploration in temporal and multivariate data. An earlier workshop publication for LFPeers included a usage scenario targeting epidemiologists and the public who want to learn from the Covid-19 pandemic and distinguish successful and ineffective measures. In this extended paper, we now show how our concept is generalized and applied by domain experts in two case studies, including a novel case on stocks data. Finally, we reflect on the new state of development and on the insights gained by the experts in the case studies on the search and exploration of temporal data to learn from peers.

1. Introduction

A common use case in many domains is the analysis of *why* entities sharing similar characteristics start to deviate from each other over time. The Covid-19 pandemic provided exactly such a use case during the first broad wave of infections in early 2020. While all countries had to contain the same virus and its mutations, there was a wide variety of policy measures, approaches, and outcomes in terms of the number of infections, deaths, and reproduction rates. This variety is not solely rooted in the political system or the economic situation of a country, as even quite similar countries in the EU had to endure more or less severe situations. Many researchers around the world have sought to understand Covid-19 indicators with various analyses, infographics, and information visualization techniques. Many approaches that include temporal analysis show *how* situations are alike and describe *how* measures have developed over time. However, few approaches also help identity and explain *why* situations have emerged, and even fewer approaches allow inferring about what we can learn from past developments of individual countries (learning from peers). Generalizable challenges include the meaningful search for similar peers based on multiple criteria, the creation and refinement of a peer set based on various data characteristics and perspectives, the exploration of temporal behavior of that peer group for multivariate time series and attributes, including the contextualization of behavior

*Corresponding author:
e-mail: sachdeva@ifi.uzh.ch (Madhav Sachdeva)
[1] ORCID: 0000-0002-9840-7735
[2] ORCID: 0000-0001-8741-9709

to explain observed effects. The complexity of data types and analysis tasks for these envisioned goals calls for visual analytics solutions.

This paper focuses on the analysis of relations within a combined dataset of temporal [1] and multivariate [2] static attributes. *To learn from peers* (LFPeers), our approach includes user-definable nearest neighbor search operations in early temporal phases, in combination with the exploration of the development of peers in later phases. We draw strong motivation from the healthcare domain, where the LFPeers concept is very relevant. Examples include public response plans [3], epidemiological approaches to attack emergent pandemia [4], epidemic management [5, 6], or the assessment of health scenarios in relation to government response events [7]. From a general, more technique-driven perspective, the LFPeers concept includes and combines approaches targeting on the *search* in temporal data based on time series similarity [8, 9, 10], as well as the *exploration* of later temporal (output) progressions [11] based on criteria such as interesting *variations* [12, 13] of peers, or *co-relations* [14, 15, 16] of peers that explain an observed variation phenomenon.

The previous version of this work [17] was initially motivated by a Covid-19 analytics usage scenario based on a dataset collected by Google [18] during the Covid-19 pandemic. We now generalize our concept and the implemented visual analytics system to provide a versatile approach for temporal similarity search and results exploration. Our principal idea is to enable users to analyze where and why a peer group of objects sharing a similar situation start to deviate from each other later in the temporal progression. We build on the user-defined similarity of observations in a search phase *before* an endpoint t_0, followed by a detailed analysis of dissimilarities of temporal progressions in an exploration phase *after* t_0. Users can define the search interval and the endpoint t_0, as it is often known when influencing factors (such as changes in policies or certain events that affect an entity) have been introduced, or interesting changes in observed measures may suggest focus points that are most useful. Pinpointing the characteristics that were different from t_0 onward helps identify variations, and helps distinguish effective from irrelevant objects or identifying irrelevant attributes for a perceived effect.

At the time of the previous work [17], none of the existing Covid-19 analytical approaches have let users control the criteria and functionality by which similarity in multivariate and time-oriented data about countries is defined. Our novel distinction between the time range before and after a focus point opened new avenues for time-oriented analysis of infection data on the per-country level. In this extended paper, we will show how this concept is generalized and applied to two different use cases in two application domains. Overall, we aim at the following contributions: first, we provide our extended concept for temporal and multivariate similarity search and exploration; second, we introduce and demonstrate the extended and generalized visual analytics system, LFPeers (short for "learning from peers"), that implements this concept; third, we show how the system is applied in an additional domain beyond the existing Covid-19 use case, evaluated in two case studies; fourth, we reflect on the new state of development and provide further ideas for future work.

2. Related Work

We first refer to conceptual and methodological baselines before we review search as well as exploration approaches, both with an emphasis on time series. Finally, we review applications that insipire the learning-from-peers concept.

2.1. Methodologies and Task Characterizations

The conceptual workflow of LFPeers builds upon principal methodologies and workflow models rooted in the visualization community, including, Shneiderman's mantra [19] and visual analytics methodologies [20, 21, 22]. Our approach differs, as we employ these overview-to-detail strategies twice: in the search and the exploration phase of LFPeers. In the *search* phase, we enable users to find a peer group of interest [23, 24]. LFPeers provides different query and filtering operations to support this search process: we consider that a query object can be known from the start [25, 26] or not [27], and that a group of peers identified through different data characteristics may be refined further through faceting and filtering [28, 29]. With *exploration*, we refer to the task of finding information, for situations where the information need is still rather ill-defined, regarding what to seek and where to seek, as characterized by Andrienko et al. [30] on spatio-temporal data exploration. With LFPeers, users explore interesting peer objects and attributes to learn from peers. With emphasis on search and exploration, the LFPeers workflow relates to the exploratory search [31, 32] methodology, enabling users to explore digital information spaces, identify query objects, and elaborate on similar objects that were retrieved. Exploratory search is applied in many domains, with digital libraries as a primary example domain [33], where the exploratory search methodology nicely aligns with the fourth paradigm in science [34]: data-driven science. Exploratory search approaches within the visualization community include approaches for movies [35], textual documents [36], semantic web data [37], World Wide Web information [38], faceted information resources [39], and text corpora [40]. In contrast to exploratory search, the learning-from-peers workflow combines an initial search phase to find relevant peers with an exploration phase downstream to explore the behavior of peers. This is why our work is strongly inspired by van Ham and Perer's "Search, Show Context, Expand on Demand" [41] methodology, as it inverts Shneiderman's mantra and the order of exploratory search.

2.2. Search in Time Series

Finding similar peers is a task applied in several fields, including social networks to find similar people, healthcare to find similar patients, or economy to find similar countries. We focus on interactive approaches for temporal data. In many time series analysis approaches, the temporal progression of an identified group of peers could provide useful insights to inform or guide decisions. Using healthcare as an example, the stratification of patients based on a similar physiology and health conditions can provide information about possible treatment strategies and future outcomes [42]. However, one challenge that search approaches have to face is to provide effective interactive visual query interfaces [31, 43] to search for temporal data [44, 45]. Two dominant paradigms are *query-by-example* (users select the query object from a pool of existing time series) [46, 45] and *query-by-sketch* (users define, sketch, draw, or create a temporal query object according to their information need) [47, 28, 44, 48, 49]. We decide for a query-by-example approach to (1) directly replay objects identified in the downstream exploration results [45], (2) not require users to know the entire data space and avoid ill-defined questions [32], and (3) ease the retrieval of exact matches [30]. Another part of the challenge in providing meaningful search support for time series is to determine time series similarity concepts that are useful for end users. The work by Gogolou et al. reveals the type of visualization influences the human similarity perception of time series. For instance, they found through a user study that horizon charts were better for similarity measures that has local variations in temporal positions than line charts and colorfields [50]. LFPeers provides useful notions of time series similarity by enabling users to define up to four configurations for time series similarity at runtime, as a basis for effective nearest neighbor retrieval and the time series are visualized as line charts and area charts. As a pre-condition, the quality [51] of time series matters, and typically there is a need for data wrangling [52, 53, 54]. We manage most of the data wrangling in an upstream pre-process, but enable users to apply data transformation and normalization operations on time series, and to define derived temporal attributes for further use [55].

In the following, we look at inspiring work in interactive similarity definition. An aspect of time series similarity is the definition of the length of the query pattern, as a basis for the online feature calculation. Here, LFPeers applies the rectangular query concept borrowed from the TimeSearcher approach [44]. Ultimately, LFPeers resolves the challenge of providing meaningful time series descriptors [56] and distance metrics [57, 58], by providing multiple choices through interactive controls.

2.3. Exploration of Time Series

A main objective of VA is to provide insights into complex, large datasets through interactive data exploration. In many approaches, an overview of the data is provided to the user to ease data exploration, and to drill down the search space with interaction techniques like zooming, browsing, or filtering. Work by Zhao et al. also introduces online data transformation of time series to assist users in deriving time series for advanced tasks and data exploration [59]. KronoMiner by Zhao et al. uses multi-foci navigation and data transformations to assist users exploring multivariate time series in multiple datasets [60]. LFPeers also supports the user in deriving time series online and is shown to generalize with two use cases. In general, visual data exploration can benefit from dimensionality reduction [61] and clustering [62] methods, to reduce the data complexity and to provide an overview of the data. For time series, examples of overviews include a clustering approach for univariate stock price time series [10], the LiveRAC [63] approach for the analysis of multivariate system management time series, and the groundbreaking calendar view by van Wijk et al. [64]. Exploration approaches combining dimensionality reduction and clustering also exist, for the visual analysis of trajectory data [65], speech intonation [66], engine data [16], horse poses [67], or embedded in data exploration workflows with provenance support [66]. LFPeers utilizes the principle of visual data aggregation, to show overviews of main characteristics of a cluster (peers), for multiple time points and multiple temporal attributes at once. Our interaction strategy in the exploration phase includes the interactive selection of an interesting reference attribute or a time point of interest, or both. In LFPeers, we formalize interestingness using two criteria. First, the variation [12, 13] of a group of observed time series (peers), to study differences in temporal behavior along the temporal progressions. Second, the correlation [15, 16] of co-existing (temporal) attributes, to identify and analyze relations [14, 55, 68] among attributes, and ultimately to explain observed behavior [69] through these related attributes.

2.4. Learning from Peers Applied

In general, LFPeers relates to approaches that are dominated by early search behavior, followed by a more open-ended data exploration phase, to learn from identified peers. From an application perspective, Covid-19 formed a strong motivator for the LFPeers approach and was also studied as a primary case in our baseline workshop paper [17] that we are extending. Related to that, Afzal et al. provide a solution based on geographical visualization to assist in building public response plans [3]. There are also several epidemiological approaches that address emergent pandemia [4] or visualizations related to epidemic management [5, 6]. Along with Covid-19, we identify the emphasis on temporal data in many approaches. The work by Leite et al. [7] is related, with their goal to support the understanding of the spread of the disease in different scenarios also in relation to different government response events, while differing in their focus on predictive modeling. LifeFlow provides overviews of event sequences for data exploration, based on data aggregations and filters [8], with an emphasis on identifying commonalities and differences across peer groups. Lifelines2 [70] stacks multiple event sequences of multiple individuals to identify variations and utilizes interactive means to determine the endpoint $t0$, similar to LFPeers where the endpoint is used to define the peer group. Users can identify effects by inferring from how other attributes related to a fixed reference attribute, even if for event sequences, not numerical time series.

Beyond the healthcare domain, van Wijk et al. [64] pioneered an approach for identifying peer groups through time series clustering in context with calendar information to reveal possible relations. PeerFinder [23] supports the task of finding similar peers based on a seed (query) object consisting of multivariate temporal and static attributes. Users can interactively refine an identified peer group, similar to LFPeers. However, exploration of peer groups and downstream explanations of relations is only implicitly supported. The Time Curves [9] and Timeseries Paths [71] approaches offer similarity-preserving layouts of time series, folded in a 2D display, allowing the exploration

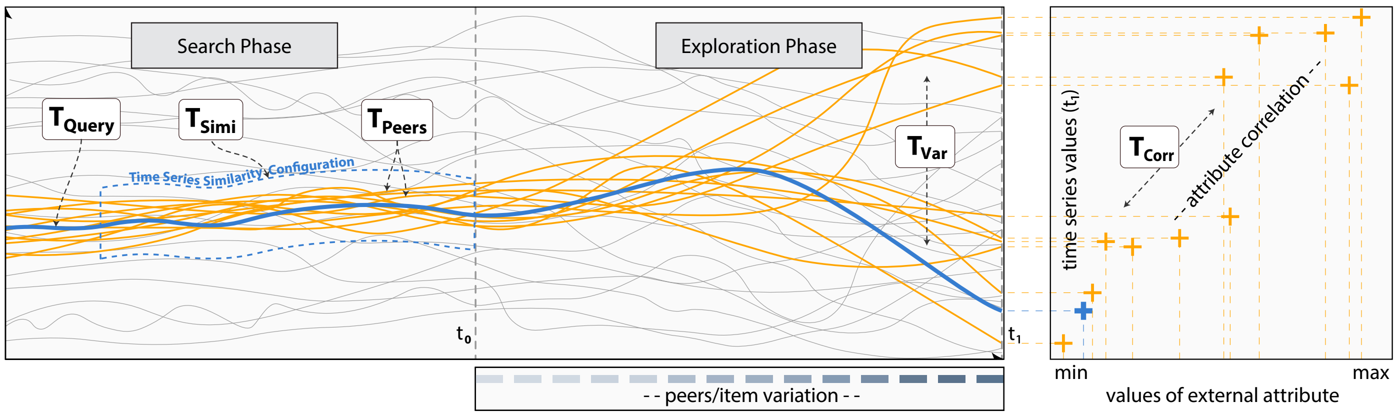

Fig. 1. Workflow with five tasks that align with the temporal progression of the data. First, users select a query object (T_{Query}) represented by its time series (blue). Second, users define similarity configurations in the search phase including a search interval and temporal endpoint t_0 (T_{Simi}). Third, based on the similarity configurations users select relevant nearest neighbor objects and apply filter operations to refine the selection of peers (T_{Peers}, orange). Fourth, users analyze variations of temporal progressions in the exploration phase from t_0 to t_1, aiming at finding dissimilar behavior (T_{Var}). Finally, users correlate values of peers of the query time series at time point t_1 (T_{Corr}) with external attributes, e.g., to identify explanations for diverging behaviors.

of variation. In contrast to LFPeers, these approaches focus on exploring self-similarity, i.e., learning from one's own past, other than from peers. Two approaches that also focus on the identification of diverging and atypical behavior of peers over time were presented by Ziegler et al. [72, 10], both with a focus on financial time series, though without peer-identification support. LFPeers also extends work by Cibulski et al., who established the identification of similar peers and the analysis of temporal observations and corresponding output progressions [11]. The approach by Stoffel et al. [73] is situated in the internet traffic domain and highlights the deviation effects from normal traffic behavior, similar to the variation analysis provided with LFPeers. Exploration of possible causes is supported by relating these behaviors to other temporal attributes showing similar behavior.

3. Methods and Abstractions

Learning from Peers. We present a design approach that supports users in learning from peer objects. In the social context, *peers* are typically people who share some commonality or behavior. We borrow from this inspiring notion and use it in a broader sense to refer to any type of data object that can be subject to analysis. As such, we explicitly include analysis cases of countries or stocks in this learning-from-peers concept. In the following, we present the problem statement that motivates our design approach, followed by abstractions of a) applicable dataset types, b) the typical workflow of users following search and exploration goals, and c) the tasks included in the proposed workflow.

3.1. Problem statement

Given a temporal, multivariate, and possibly multimodal collection of data objects, the goal of analysts is to identify peer objects with joint behavior early in time, and to explain variations of (unexpected) behavior later in time. This learning-from-peers concept allows users to make informed decisions. One of the driving examples was the Covid-19 crisis that motivated the following analysis question: "what can a country learn from its peers when, at some point in time, these peers started improving, e.g., with respect to lower numbers of new confirmed cases?" An example derived from stock data analysis is: "given a stock and its peer group with similar performance, why did some peers start performing increasingly better?"

Inspired by the idea to *learn from peers* and possibly do better in the future, most challenging aspects in the analysis workflow are to identify peer objects with joint behavior in a dataset with temporal and multivariate complexity, to compare these objects over time, and to explain variation between these objects, observed in later temporal progressions. The analysis problem is special, as analysts face the challenge of an interesting temporal duality. In an early phase of the time series, the challenge is to identify and select a subset of *most similar objects*. This search process can be complex due to different temporal attributes that may contribute to the overall similarity definition, as well as possible object faceting and filtering challenges to arrive at a useful peer group. In contrast, in later phases of the time series, the analysis question inverts toward the exploratory *identification and explanation of variation (dissimilarity)* between previously selected peers. The challenge in the exploration phase downstream is to identify, assess, and possibly explain these variations discovered in the peer group, by considering the multivariate nature of the data in the learning-from-peers process.

3.2. Design Approach

The typical learning-from-peers workflow of users consists of a search and an exploration phase, which we further break down into five analysis tasks. The primary data type of analysis is time-oriented data, which has a methodological implication on the analysis design. On the one hand, we align the early search phase in the workflow with an early interval within time-oriented data. On the other hand, users conduct the later exploration phase at a later interval within the time-oriented data. As such, this *process-oriented perspective of the workflow* and the *time-oriented perspective of the data* at hand align.

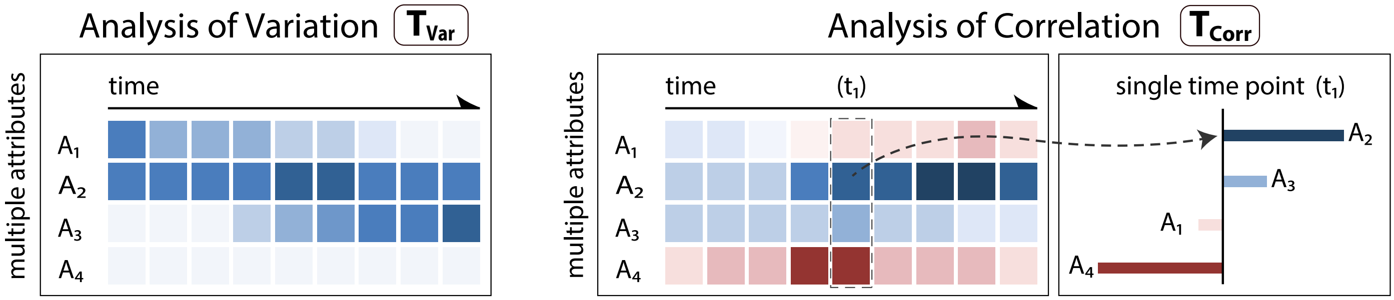

Fig. 2. Extension of the analysis for variation ($\mathbf{T}_{\text{Var}}$) and correlation ($\mathbf{T}_{\text{Corr}}$) for multi-attribute analyses, to further support analytical reasoning. In the exploration phase, users are tasked with the assessment of a temporal reference attribute. With the multi-attribute support, users can identify interesting attributes more easily. Either to identify other attributes with interesting variations across peers ($\mathbf{T}_{\text{Var}}$, between-objects focus), or to identify correlated attributes that explain these variations ($\mathbf{T}_{\text{Corr}}$, between-attributes focus). On the right, we illustrate the detailed focus on correlations for single time points.

3.3. Abstractions

We use the time axis to structure the presentation of these abstractions, as the evolution of the data over time nicely aligns with the order of phases in the workflow of analysts. Accordingly, we structure abstractions across the temporal domain of the dataset (see Figure 1).

Data abstraction. Across the entire data analysis process, analysts focus on the very same type of data objects (countries in the Covid-19 case, stocks in the finance data case) which are represented by multiple attributes. Mandatory for this type of approach is the existence of at least one time-oriented attribute, as the goal of the analyses is related to the temporal analysis of the data, i.e., the temporal progression of one or multiple temporal attributes. In Figure 1, temporal data of all objects is visualized with light gray color as a background layer, for a reference attribute. What makes the approach special is the existence of additional data attributes, which may or may not develop over time. These additional data attributes will help to contextualize or even explain findings in the temporal reference attribute. It may be desirable to include different and even multivariate sources of data into the approach [2], inspired by the idea to increase the variety of interesting connections [74] that can be drawn across attributes. Using the Covid-19 case as an example, we included data from various domains (epidemiology, government responses, mobility, etc.) into LFPeers, all linked to the data object at hand (countries) using a primary key concept. For the stocks dataset, we include basic information about companies, quarterly report attributes, and the stock prices.

User Workflow and Analysis Tasks. The workflow of analysts includes two phases. In an early *search phase*, users retrieve peer candidates within a user-defined temporal interval prior to an endpoint t_0 (see Figure 1). The search for peer candidates may be based on a nearest neighbor retrieval [75] for one temporal attribute, or include multiple temporal attributes serving as similarity concepts. The group of peers may be synthesized from these complementing retrieval results. Also, other criteria in the dataset may influence the definition and refinement of the peer group. Examples include static data attributes like the gender of persons or the geographic region for country objects, or the structural linkage of objects provided through object-connectivity information. The *exploration phase* starts after the search phase, i.e., after endpoint t_0. In this phase, variations between peer objects may occur: time series of the peers that have been quite similar prior to t_0 are starting to fan out and behave/perform differently. Users identify these variations in the peer group, and analyze correlations to identify attributes that may help to explain these variations. In the exploration phase, users may want to focus on a second endpoint t_1, a time point where an observed pattern of variation can be analyzed in detail most effectively. For the design of visual analytics support, we suggest the set of five abstracted tasks, as presented in Table1.

We refine and extend [17] the task characterization for the exploration phase, to better distinguish visualization design targets for a single-attribute or a multi-attribute focus. This applies to both variation and correlation analysis. Figure 2 provides a conceptual overview of the novel multi-attribute support.

4. Non-Visual Support

In this section, we present the four main types of data transformations and algorithmic models used in LFPeers. Based on the abstractions made on data and tasks (Section 3.3), we describe the online data transformation (Section 4.1), online feature extraction (Section 4.2), the use of machine learning support through clustering and dimensionality reduction (Section 4.3), as well as further exploration support (Section 4.4).

4.1. Online-Data Transformation

With LFPeers, users are in control of the similarity configuration used to identify peer candidates ($\mathbf{T}_{\text{Simi}}$). Upon interactive change, a data transformation and feature extraction pipeline is executed, including the following steps.

Derived Time Series. Users can create temporal secondary data, derived from original time series. With this approach, the original time series itself remains untouched, to serve as a basis for multiple similarity configurations. Users can choose from a set of processing operations, including smoothing, normalization, derivation, discretization and logarithmic filters, or scaling. The data syntax of all these operations is the same, having a time series as input and a time series as output. We utilize this

Task	Phase	Description
$\mathbf{T}_{\text{Query}}$	Search	select query object of interest. In case of a clear information need, this action is straightforward. In turn, the identification of a query object can also be the result of an insight gained through data analysis.
$\mathbf{T}_{\text{Simi}}$	Search	define similarity configurations and search time intervals. In this way, algorithmic support can be utilized to retrieve meaningful peer candidates. As the user's notion of time series similarity may be multi-faceted, multiple similarity configurations would be useful. Similarity configurations include a search time interval and an endpoint t_0, which will form the temporal transition between the search and the exploration phase.
$\mathbf{T}_{\text{Peers}}$	Search	identify, select, and refine group of peers with most similar behavior, prior to endpoint t_0. In this way, user's determine a final peer group based on different criteria, for the exploration phase downstream.
$\mathbf{T}_{\text{Var}}$	Explore	explore variation among peers, to identify commonalities and differences between peers for one attribute. This analysis can be to one reference attribute, or be extended to other temporal attributes of the multivariate dataset.
$\mathbf{T}_{\text{Corr}}$	Explore	explore correlations between a reference attribute of interest and external attributes (factors) in the dataset, that may explain variations in the post-progression of peers after t_0. Formally, correlation analysis includes value distributions of two attributes for a given time stamp, and can be applied to a predefined attribute pair $(1 : 1)$, or be applied to a reference attribute and all remaining attributes $(1 : n - 1)$.

Table 1. Task abstraction, along the search and exploration phase (cf. Figure 1). Explore tasks can be applied for single or multiple attributes (cf. Figure 2).

by allowing users to apply these operations in arbitrary order, to guarantee maximum flexibility in data processing.

Derived Data Types. In some cases, it makes sense to transform time series into a different data type, to make the data more useful for the users' intended similarity task. LFPeers supports turning temporal attributes into event sequences, by identifying time points of interest such as minimum, maximum or threshold crossings. Also, temporal attributes can be reduced to static attributes, by mapping the temporal values to statistical values, such as value ranges or number of events.

4.2. Online-Feature Extraction and Similarity Calculation

Time Series Descriptors. With the interactive definition of a similarity configuration ($\mathbf{T}_{\text{Simi}}$), LFPeers computes feature vectors from the targeted temporal attribute. Users can choose between different time series descriptors, to transform the temporal data into numerical feature vectors. The syntax of the descriptor software interface expects a collection of time series as input, and produces a collection of numerical multivariate feature vectors as output. Supported implementations range from equidistant temporal sampling to features of statistical measures including min, max, and trend values. The set of descriptors is extendable, e.g., by PAA [76] and PIP [10] algorithms.

Distance Metrics. Users can also choose between distance metrics, to be applied to the feature vectors calculated from the time series, or other derived data types. Default implementations for the temporal features include the Euclidean distance, if the exact shape and temporal occurrences shall be emphasized [77], as well as dynamic time warping [78] if similar structures shall be considered with a certain tolerance in temporal distortion. Also, LFPeers supports single-value computations for (derived) static numerical attributes, such as min, max or difference measures. These are useful if the user wants to query for trends or the extent of a temporal development within the selected time range. Finally, the distance between event sequences is computed via sequence alignment based on the number of events within the time range.

4.3. Unsupervised Machine Learning Support

Besides search and retrieval support, LFPeers supports users in identifying and refining group of peers ($\mathbf{T}_{\text{Peers}}$) as a baseline for the exploration of peer behavior down the line. At a glance, LFPeers provides visual clustering and dimensionality reduction methods, to better exploit structural characteristics of the temporal data represented through multivariate features (cf. Section 4.2).

Clustering and Grouping-Based Coloring. The default clustering method of LFPeers is k-Means, as a fast and well-known variant that enables users to control the number of clusters interactively. Other clustering implementations can be added to the approach, following the syntax of having multivariate feature vectors as input, and a set of a collection of objects as output. We combine clustering with a grouping-based coloring, to have a group-structure-preserving coloring on demand.

Dimensionality Reduction and Similarity-Based Coloring. LFPeers includes several representative dimensionality reduction methods [79] (PCA, t-SNE [80], UMAP [81]). Alternative implementations can be added, given that they have numerical feature vectors as input and a low-dimensional numerical representation as output. The output of methods is always two-dimensional, serving two purposes. First, the output serves as the natural input data for scatterplots used in LFPeers. Second, we apply similarity-preserving 2D colormaps [82] to link projected objects across views. We decided for a cube-diagonal RGB 2D colormap [83], based on its high-color exploitation and acceptable perceptual linearity preservation.

4.4. Exploration Support

Users need the means to assess the time-varying behavior of peers for every temporal attribute provided in the dataset. LFPeers provides automatic support to help users identify varying behavior quickly, in combination with visual interfaces serving as visual cues showing interesting temporal behavior. We support the two exploration tasks as follows:

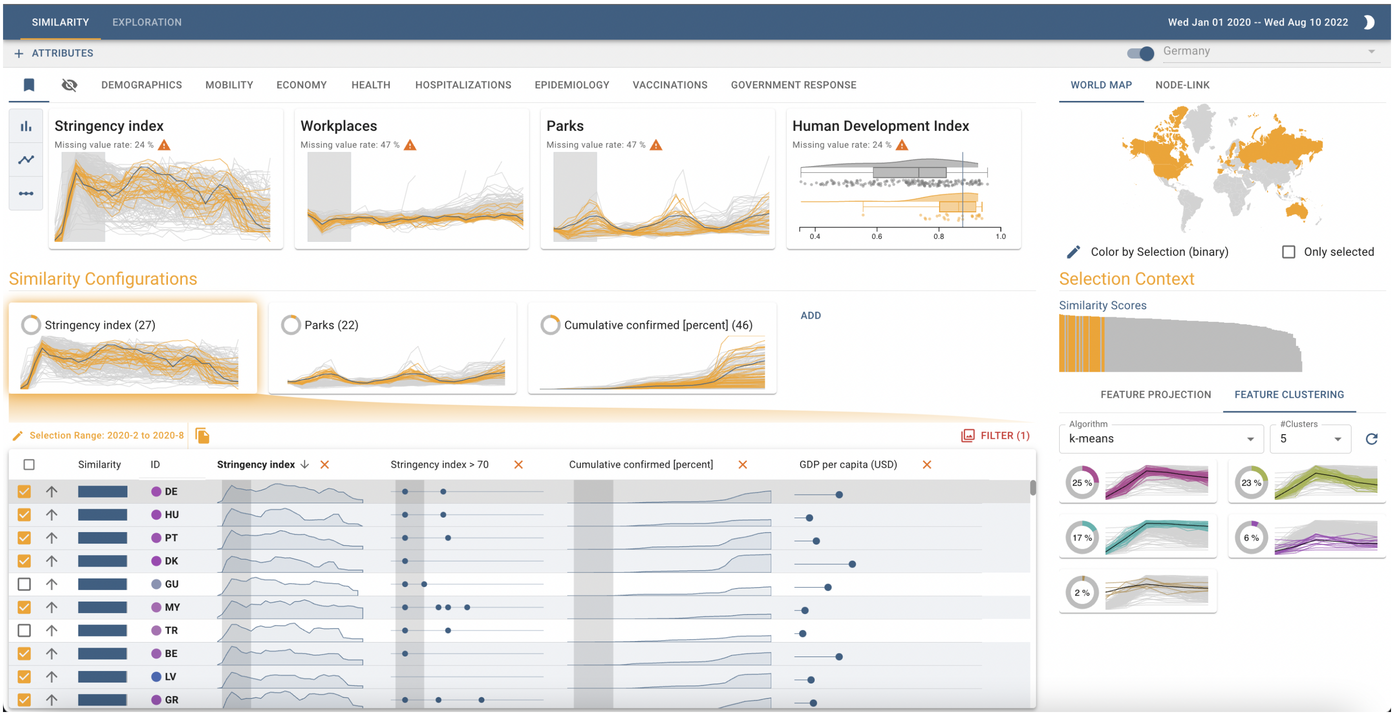

Fig. 3. The search interface of LFPeers with the Attribute Panel at the top left, Similarity Configurations View, and the Similarity Configuration Table below. In the example, three similarity configurations have been defined, currently active is the configuration based on the *Stringency index*. The user has placed several attributes in the table, including the temporal query attribute (which currently drives the sort order), and, e.g., a derived event sequence (showing when the *Stringency index* exceeded a threshold of 70). The right side shows the World Map of selected countries, the similarity distribution, and five identified clusters.

Analysis of Variation. With $\mathbf{T}_{\text{Var}}$, users can identify variations of the temporal progression *within a single* temporal attribute. The algorithmic support is designed to highlight fan effects in temporal attributes, i.e., effects where identified peers show increasingly diverse and varying behavior during the exploration phase. The input for this non-visual support is the value distribution of a (temporal) numerical attribute at a given time point, the output is a positive numeric value representing the attribute variation, using a *unipolar value range*. As a default implementation, we use the standard deviation.

Analysis of Correlation. With $\mathbf{T}_{\text{Corr}}$, users can identify relations of the temporal progressions *between two* temporal attributes. Here, visual cues in LFPeers will highlight the distribution of any two given (temporal) attributes that are strongly correlated, either positively or negatively. As such, the input for this non-visual support is the value distributions of two (temporal) numerical attributes at a given time point, the output is a numeric value representing the attribute correlation, using a *bipolar value range*. As a default implementation, we use the Person correlation measure, as an example for linear correlation. The Spearman rank correlation measure would be an alternative.

5. The Visual Search and Exploration Interfaces

LFPeers is a visual analytics system that adopts the conceptual workflow (Figure 1) in two dedicated interfaces, for the search phase (Figure 3) and the exploration phase (Figure 7).

General Design Decisions. Following the conceptual workflow in Figure 1, LFPeers applies a common color coding across views. Time series are represented with bundled line charts [84, 85], with the selected object ($\mathbf{T}_{\text{Query}}$) drawn with a gradient blue-green line always on top. The gradient coloring mirrors the y-axis position and helps to distinguish the trends of the stacked spark lines within the dense table layout. Orange color is reserved for selected peers ($\mathbf{T}_{\text{Peers}}$), here to support superimposed visual comparisons [86] with the selected object. All other objects are drawn in gray to always provide the global data context. Non-temporal attributes are presented as raincloud plots [87] (numeric), or bar charts (categorical). The corresponding views are linked and update upon user interaction, changing color to distinguish the base distribution (gray) from the current group of peers (orange).

5.1. Search Phase

System Overview. The search interface (Figure 3) provides an overview of all data objects, with an emphasis on temporal and static attributes. Users can browse through distribution plots of attributes, and drop interesting attributes into a sortable table for a closer, comparative view, as a baseline for querying similar objects, as well as defining and refining a selection of peers. The right side of the interface provides a map component, as well as the results of dimensionality reduction and clustering algorithms, to provide additional perspectives on the data from a structural point of view.

5.1.1. Selection of Query Objects ($\mathbf{T}_{Query}$)

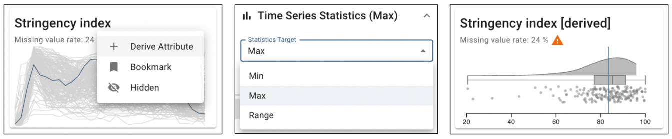

Fig. 4. Interactive attribute processing. Here, the user derives a static attribute using the maximum function for the *Stringency Index* attribute.

An early task in the workflow of users is selecting a query object. The query object will be used to determine the group of similar peers, to learn from. In case users have a clear information need from the start, a combo-box at the top right allows the straightforward selection of a query object. The example in Figure 3 uses *Germany* as the country in the case study that wants to learn from peers how to cope with the Covid-19 situation. Query objects can also be identified through data analysis. Whenever a single object is selected through brushing a line in line charts or point in scatter plots, the query object is automatically set. Similarly, a click on a row (object) in the table, and on the map (if objects refer to geographic regions like countries), defines the query object. Fixing the query object is possible through a slider control at the top right, close to the combo box.

5.1.2. Similarity Configuration and Querying ($\mathbf{T}_{Simi}$)

LFPeers allows the analysis of multiple attributes at once, to identify temporal attributes of interest relevant for further analyses. The analysis task related to attribute-based analysis is the definition of similarity configurations, based on individual attributes of interest ($\mathbf{T}_{\text{Simi}}$). To start with, users can select between attribute categories at the top left of LFPeers, e.g., if multiple sources of (temporal) attributes are included and combined in the approach. In the example shown in Figure 3, eight sources of attributes are provided with different tabs, starting with *Demographics* on the left, and ending with *Government Response* on the right.

Attributes Panel. The upper left panel also allows users to browse through a selected group of attributes and analyze details of attribute distributions. In line with the general design decisions, bundled line charts show the distributions of temporal attributes, raincloud plots and bar charts are used for numeric and categorical non-temporal attributes. Also, charts for the visualization of attributes contain an orange glyph icon to indicate missing values. If users want to transform, process, or derive attributes, a right-click on an attribute opens a pop-up menu, with controls to select and parameterize attribute-based operations. Figure 4 provides an example.

Similarity Configuration and Selection. The *Similarity Selection Panel* below the *Attribute Panel* on the left (cf. Figure 3), helps users to orchestrate similarity configurations. In the example, three configurations have been created. The similarity configuration of the *Stringency Index* attribute is currently selected, allowing the refinement of the configuration through interactions as follows. Users can define the time interval that is used by LFPeers to compare time series and retrieve nearest neighbors. The end of this time interval also defines t_0, the endpoint of the search phase in the time series. Inspired by the timebox widgets approach [44], users can brush any temporal chart to define this time interval. Other interactive manipulations of the similarity configurations include data transformations applied to the temporal attribute. Here, users can choose between (combinations of) smoothing, normalization, derivation, discretization, and scaling functions (cf. Section 4.1) through a context menu popping up on request (right-click on a temporal attribute). Users can also control the feature extraction method and distance metric (cf. Section 4.2). For dataset sizes below 1,000 objects and time intervals below 100 time points, the online computation of a similarity configuration is possible within milliseconds to seconds, depending on the data processing and algorithm choices made. Besides temporal attributes, users can also create similarity configurations for derived event sequences, single-value derivatives, and other static numerical or static categorical attributes (by using context menus).

Similarity Configuration Table. A table in the search interface shows attributes of interest as columns and retrieved objects as rows, Figure 3 shows an example at the lower left. Attributes can be added from the *Attribute Panel* via drag & drop, which also triggers the similarity configuration process. Objects in the table are the results of query operations applied to the similarity configurations. Temporal attributes are displayed as spark lines, static categorical attributes as labeled color-coded rectangles, numeric attributes as lollipop markers and event sequences as points on a timeline. Sorting of objects by a temporal or static attribute is supported by two criteria: by (average) value, or by similarity to the query object. If such a query object has already been selected, it is visualized as a fixed top row, with the similarity-based sorting criterion as default, to naturally support the analysis of retrieved nearest neighbor objects. Individual time intervals of the similarity configurations are always shown as gray background areas, across all temporal attributes, as shown in Figure 3 for the *Stringency index* and the *Cumulative confirmed [percent]* attributes.

5.1.3. Identification of Peers ($\mathbf{T}_{Peers}$)

Multiple data characteristics exist that may influence users in identifying, selecting and refining the group of peers to learn from $\mathbf{T}_{\text{Peers}}$). Individual temporal attributes are among the strongest source of information to perform this type of temporal analysis. Alternative data characteristics that can be leveraged with LFPeers are derived event sequences, static attributes, and features of temporal attributes. In addition, LFPeers provides peer-selection support for optional geographic or object-connectivity (graph) information (cf. Section 3.3).

A peer group is always empty at start. Users can make use of various views provided with the search interface (cf. Figure 3), to analyze, extend, and refine the peer group from different perspectives on the data. These views are linked, i.e., any change to the peer group is automatically propagated to all corresponding views, always showing peers in orange color. This highlighting of selected peers also enables users identifying and removing outliers, as a means to fine-tune the peer group. We describe the peer identification process along the lines of supported data characteristics.

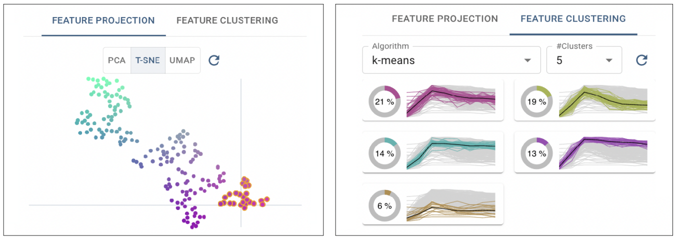

Fig. 5. Structural characteristics of features can be analyzed using dimensionality reduction and visual clustering. Scatter plots show the result of dimensionality reduction (left), bundle line charts show clustering results (right). Both the dimensionality reduction and the clustering can be used to create a color coding (continuous vs. discrete) that is linked across views.

Peer Selection Based on Temporal Attribute. One way of identifying the peer group is using the ordered list of data objects in the *Similarity Configuration Table*. The table shows retrieved objects according to the similarity configuration of a selected attribute, and allows the detailed analysis comparison of retrieved objects. A straightforward approach may, e.g., be to add the first k neighbor objects of a single similarity configuration. An additional similarity column in the table shows object similarities as blue horizontal bars, easing the selection of peer objects and the identification of similarity thresholds. In the table, a checkbox control allows adding objects to the peer group.

Peer Selection Based on Multiple Temporal Criteria. To further support forming peer groups based on multiple criteria of similarity, LFPeers supports the parallel definition and analysis of up to four similarity concepts. Each concept can be based on entirely different attributes and definitions of similarity, leading to different collections of retrieved data objects. Both the *Similarity Selection Panel* and the *Similarity Configuration Table* provide perspectives on the temporal attributes, and on the retrieval results. Figure 3 provides an example on the lower left, where the *Stringency Index* similarity concept is currently active. With the table and checkboxes per object, users can select and refine the final peer group, based on the multiple similarity criteria. In addition to temporal attributes, users can also incorporate the retrieval results of derived event sequences, as it can be seen in Figure 3 for the *Stringency Index* attribute. Also, static attributes can form the basis for similarity configurations, as it can be seen for the *GDP per capita (USD)* attribute in the *Similarity Configuration Table* in Figure 3.

Peer Selection Based on Features. The multivariate features computed through the similarity configurations can reveal additional structural characteristics of the data. LFPeers supports feature-based analysis by offering visualizations of dimensionality reduction and clustering results (cf. Section 4.2). These auxiliary views are located on the lower right of the search interface (cf. Figure 3). Users can switch between the results of PCA, t-SNE and UMAP by using a scatter plot. Each object is mapped into the numerical 2D space, with the default color scheme being applied (blue for the query object, orange for peers, gray for remaining objects). As an alternative, users can also link the position information across views, by applying a similarity-preserving 2D colormap (cf. Section 4.2). A

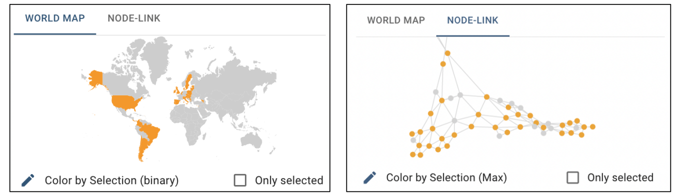

Fig. 6. Optional perspective on the data, for geographical or object connectivity information. The World Map allows the analysis of the distribution of the peer group around the globe. The node-link diagram supports the analysis of connected objects. Here, the peer group color coding is used.

crosshair additionally marks the query object, allowing a more fine-grained assessment of neighborship relations. Lasso-based brushing allows the interactive refinement of the peer group. As an alternative, users can analyze groups of objects by studying the clustering result, as shown in Figure 3. Each identified cluster is displayed with a size indicator using a donut chart, and the distribution of the time series is visualized as line bundles. Additionally, we draw the centroid time series on top, for each cluster. Clicking and brushing selection allows users making changes to the peer group based on insights gained through visual clustering.

Peer Selection Based on Optional Data Characteristics. LFPeers currently provides perspectives on two other data characteristics that may apply to the cases at hand: geographical and connectivity information (cf. Section 3.3), displayed at the upper right of the search interface, shown in Figure 6 in detail. Both perspectives form the basis for peer identification, selection, and refinement. For datasets with geographic information, a *World Map* (Figure 6) displays how data objects are distributed across the globe. This can be useful to identify whether the selection is wildly scattered across the world, or grouped in specific regions. Users can identify outliers, such as countries notable missing from a dense cluster, or far away on other parts of the world. The map is linked to the global color scheme and automatically responds to change, accordingly. The user can click on countries to alter the selection of peers. Naturally, a $1:1$ relation between a country and an object may exist. As an alternative, LFPeers also supports $1:n$ relations, if a selected country on the map represents multiple objects, such as local regions or counties. LFPeers then additionally displays the number of objects per visualized country. For datasets with connectivity information, structural linkage between data objects can be visualized with a *node-link diagram* (Figure 6).

Data objects are represented as nodes of the graph, which are placed by a force-based layout algorithm. With this graph visualization, users can identify groups with strong connectivity, which can be useful to further refine the peer group based on relationship structures. Again, the object colors are coordinated with the global color scheme.

5.2. Exploration Phase

The exploration interface (Figure 7) is designed to support users in the exploration of the temporal development of the peer group defined in the search phase. At a glance, the interface is divided into two main components: At the top, an auxiliary

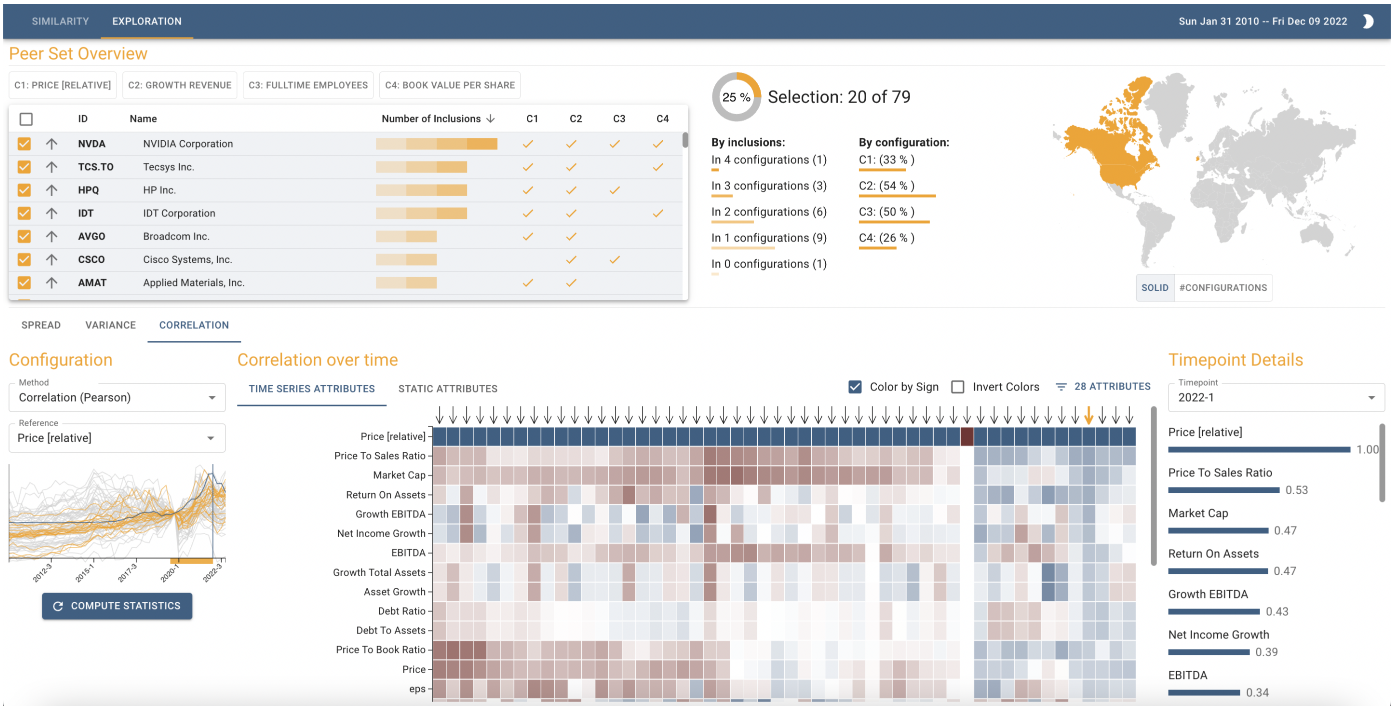

Fig. 7. The exploration interface of LFPeers. In the upper section, users have an overview of the data characteristics of the defined peer group, as defined in the search phase. The overview is interactive and allows the refinement of the peer group ($\mathbf{T}_{Peers}$), based on insights gained in the exploration phase. The lower section of the exploration interface provides information about the variation behavior of peers in temporal attributes ($\mathbf{T}_{Var}$), and correlations to a reference attribute ($\mathbf{T}_{Corr}$). Visual cues show variation and correlation statistics for every attribute and time point, and guide users towards interesting patterns to learn from. Here, The Price [relative] attribute of stocks is selected as reference attribute, and attribute correlations are revealed in the matrix.

overview of the main characteristics of the final peer group, and at the bottom the core views for the variation and correlation analysis components. Here, users can switch between variation analysis and correlation analysis through a control on the lower left. According to the general design decisions, the selection color for objects of the peer group is always orange, while the gray color represents the remaining dataset as a whole.

Peer Group Overview and Refinement ($\mathbf{T}_{Peers}$). For the users' convenience, the exploration interface displays main characteristics of the peer group defined in the search phase, and allows the refinement of peers. We have decided for this redundant support of $\mathbf{T}_{Peers}$ in both phases, to mediate between the search phase and the exploration phase, without the need of users to switch between the two interfaces. The overview table on the left provides an overview of the peer selection, including all similarity configurations made (C1-C4), as Figure 7 shows, for the stocks analysis case. Users can sort by each column, and make refinements to the peer group on demand. The peer group overview at the top is complemented by a statistics panel at the center and a World Map on the right. The statistics panel provides information about the size of the selection and the containment of peers with respect to the four similarity configurations. The World Map helps to understand the spatial distribution of peers (countries) in detail and can be used to refine the selection further.

5.3. Exploration of Attribute Variation ($\mathbf{T}_{Var}$)

At the heart of the learning-from-peers concept is searching for a similar peer group in an early temporal phase, and to identify diverging behavior between peers in a later exploration phase. LFPeers supports the analysis of variation between peers ($\mathbf{T}_{Var}$) at two granularities: for multiple temporal attributes, and a single temporal attribute.

Variation Analysis for Multiple Attributes. The Temporal Heatmap View serves as the starting point for the exploration of variation between objects of the peer group. The heatmap shows variation information of multiple attributes and multiple time points, thus having the highest information density in the exploration process. The heatmap shows the statistical variation values, computed through the non-visual support (cf. Section 4.4). We use the brightness channel to show the values, with high variation values having blue colors. Figure 8 (left) shows an example for 12 attributes and 32 aggregated time points. Users can identify temporal variation patterns along the x-axis (time) for multiple attributes at a glance. The various attributes are sortable by variation per time point, to further support the between-attribute comparison. In Figure 8, the user sorted by the 7th time point, bringing the *Transit stations* attribute, having the highest variation for this time point, to the top. In general, the analysis of variation in the Temporal Heatmap View can reveal two main types of insight. Primarily, users can identify an attribute with interesting variation, and select it to define it as the reference attribute downstream. With a single attribute at hand, users can perform detailed analyses, e.g., compare the query object with other peers (see below).

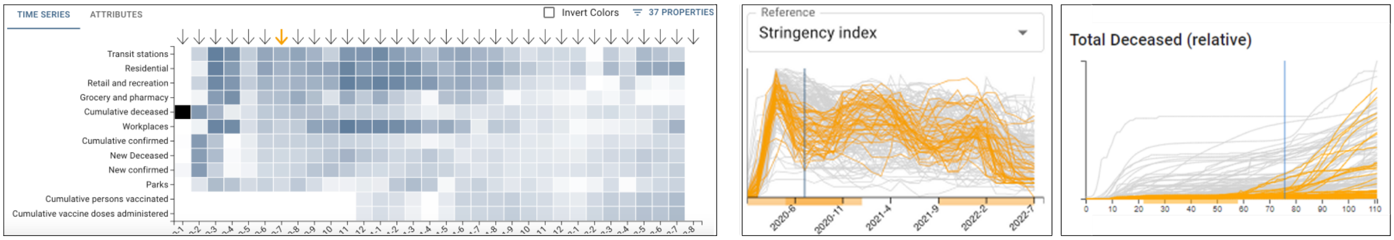

Fig. 8. Two views for the analysis of variation ($\mathbf{T}_{\mathrm{Var}}$): the Temporal Heatmap View for multiple attributes (left) and bundle line charts for single attributes (right). The typical user workflow supports users in identifying an interesting reference attribute with the heatmap, and analyzing this focused reference attribute in detail with a bundle line chart.

Alternatively, the heatmap can be used to identify interesting patterns between attributes, which forms an informed start for correlation analyses (cf. Section 5.4).

Variation Analysis for a Single Reference Attribute. There may be an attribute of general interest, and attribute of the similarity configurations, or an attribute identified in the Temporal Heatmap View that gains the attention of the user. With the bundle line chart visualization in the lower part of the exploration interface, users can analyze the temporal behavior of the peer group for a single attribute in detail. Users can analyze the temporal development of peers at high resolution, in contrast to the heatmap that shows a discretization of temporal variations. Of particular interest is the comparison of variation between the blue query object defined in the search phase ($\mathbf{T}_{\mathrm{Query}}$) and the temporal progressions of the orange peer group ($\mathbf{T}_{\mathrm{Peers}}$), both can be seen in the lower left line bundle chart (cf. Figure7).

5.4. Exploration of Attribute Correlation ($\mathbf{T}_{Corr}$)

Correlation analysis complements the within-attribute analysis of variation by the between-attribute analysis of correlation. Relations between attributes are useful cues to contextualize, describe, and explain identified behavior of peers.

Correlation Analysis for Multiple Attributes. The non-visual support of LFPeers (cf. Section 4.4) provides one correlation value per timestamp, for any attribute other than the user-defined reference attribute. The reference attribute is defined upfront, e.g., through the analysis of attribute variation with $\mathbf{T}_{\mathrm{Var}}$. The exploration interface can be switched to correlation analysis, and the Temporal Heatmap View automatically adapts to the visual representation of correlations between the reference attribute and $n - 1$ remaining attributes. Figure 9 (left) provides an example with 14 attributes. The bipolar nature of correlation values is reflected by blue colors for positive and red colors for negative correlations, again encoded with the brightness channel. As before, users can sort the heatmap by time stamp to get different perspectives on temporal attribute correlations. The insights gained with the Temporal Heatmap View allow proceeding with the workflow in two directions (both described in the following): First, users can focus on single time stamps to analyze correlation values in detail for all attributes. Second, users can put emphasis on an identified attribute pair, to analyze attribute correlations over time in detail.

If users have identified and selected a single time point of interest, LFPeers provides a view that helps users identifying (multiple) external attributes that help explain some temporal phenomenon identified earlier in the reference attribute. To support this task with automatic guidance, a list-based view ranks attributes by the strength of the correlations to the reference attribute. The ranking uses absolute values, to also account for negative correlations, with the strongest absolute correlations across attributes always being presented on top. Blue bars in combination with the display of the strength of the correlation ease the analysis, red bars encode negative correlations. In the list, the selection of an attribute of interest facilitates the information drill-down to the most detailed correlation analysis for an attribute pair.

Correlation Analysis for an Attribute Pair. With the reference attribute and an identified (correlated) attribute at hand, users can analyze the temporal developments of the peers' temporal correlations in detail. Figure 9 (right) provides an example, showing two bundled line charts (of the *Price [relative]* and the *Book Value Per Share* attributes) as well as a scatter plot on the right. The two bundled line charts allow the detailed temporal assessment and comparison, while the scatter plot shows the correlation between the two attributes for a single time point. The latter is the most detailed view provided with the exploration interface, facilitating the analysis of user-selected attributes and a user-selected time point.

6. Implementation

LFPeers is implemented as a single-page web application using the React framework. Interactive visualizations are rendered as SVG elements using D3.js, those with a larger number of visual elements use custom implementations on canvas elements. On the backend side, a PostgreSQL database, Node.js and FastAPI services provide data storage, -aggregation, -streaming and computation of statistics and projections. All similarity and sorting computations are run on the client to ensure high responsiveness, essential to create a user experience of rapid exploration and feedback in the search phase. The whole system is deployed as a bundle of docker images.

7. Case Studies

We present two case studies conducted with domain experts with real-world datasets reflecting their domain expertise. The

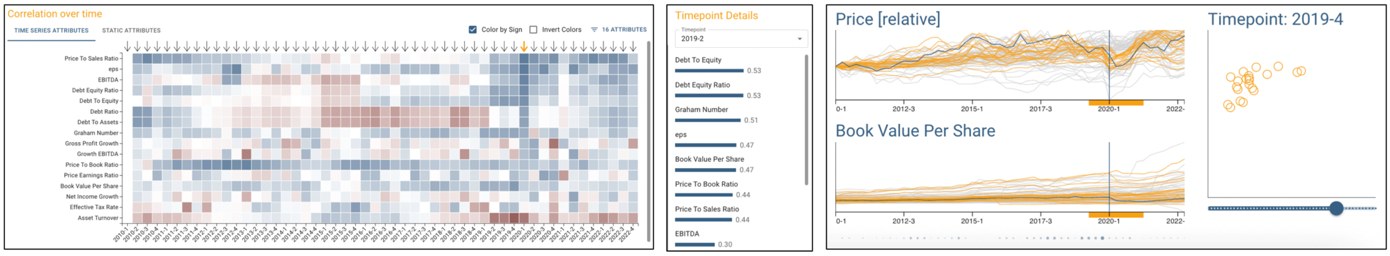

Fig. 9. Two views for the analysis of correlation ($\mathbf{T}_{\mathrm{Corr}}$): the Temporal Heatmap View shows correlation values between a reference attribute and multiple attributes (left). Users can analyze details of a single time point for all attributes, and find most correlated attributes in a list bases view (center). The datailed analysis of two (correlated) attributes over time with two bundle line charts and a correlation scatterplot (right) reveals relations between attributes and helps explain variations identified in the reference attribute earlier, here, between the Price [relative] and the Book Value Per Share attributes of the stocks dataset.

datasets were provided by the authors. The sessions were conducted remotely via a video-conferencing tool, and the experts and interacted with LFPeers using the remote control on their own devices. In both cases, we first introduced the experts to the conceptual idea of learning from peers, gave an introduction to the two interfaces of LFPeers and explained its main interactive capabilities. Next, we asked the experts to speak aloud while conducting their case along the lines of the five tasks, while we observed the expert, to protocol the workflow. Each use case lasted approximately 1 hour.

While presenting the two case studies, we refer to figures in the supplemental materials document, to be precise about the actions taken and insights gained by the experts. The case studies describe the actual sequence of actions performed by the experts. When referring to figures, we use the following codes $SUP.Figx$, where x refers to the figure number in the supplemental materials. An overview of the Covid-19 case study is provided with Figure 10, Figure 11 does the same for the stocks data case.

7.1. Case Study: Covid-19

In this case study, we collaborate with Christina, a healthcare researcher in the Epidemiology Department of the University of Zurich, working with Covid-19 data. As a first action, Christina familiarizes herself with the different types of attributes, contained in the eight different attribute categories tabs at the top left of the search interface of LFPeers (*SUP-Fig1*). She identifies the *Cumulative confirmed* attribute and wants to use it for similarity search. However, the attribute has absolute values and only very few countries are visible in the bundle line chart, whereas most countries have values close to zero (*SUP-Fig1*, top left). From a set of transformations (*SUP-Fig2*), she decides for the scale transformation to derive a new attribute that is scaled relative to a country's population (*SUP-Fig3*). The execution of the attribute transformation provides the first similarity configuration ($\mathbf{T}_{\mathrm{Simi}}$), automatically applied on this *Cumulative confirmed [relative]* attribute. As a next step, in the search interface, she adds the *vaccination policy* attribute as a second similarity configuration, visible in the Similarity Configuration View and the Similarity Configuration Table, both on the lower left of (*SUP-Fig4*). She identifies that many countries have a staircase pattern in the six-month time range she selected, and Christina decides to choose the United Kingdom (GB) as the query object ($\mathbf{T}_{\mathrm{Query}}$, representative of this pattern (*SUP-Fig4* top right, in the combo box). She also finds out that some of the countries currently in the peer group are distributed across the globe, by inspecting the Map View. She decides to focus on European countries only, following the idea that for the search for peers in the Covid-19 case, countries closer to the United Kingdom (the query object) would provide a more representative peer group. In (*SUP-Fig5*), the filter interface can be seen, used by Christina to restrict the peer definition to European countries. After the definition of a peer group in the search phase using the two similarity configurations and the filter ($\mathbf{T}_{\mathrm{Peer}}$), Christina transitions to the exploration interface.

In (*SUP-Fig6*), Christina re-identifies the peer group defined so far and further limits it to the 11 peers that match both similarity configurations well (C1 and C2 in the top left table view). The center of the view at the top shows statistical information about the peer group, the Map View on the right validates the European-only selection. Next, she starts with a correlation analysis ($\mathbf{T}_{\mathrm{Corr}}$), having defined *Cumulative confirmed [relative]* as the reference attribute (*SUP-Fig6*, lower left). By looking at the bundle line chart on the lower left, she identifies a steep increase of many peer countries in January 2022. This visual cue made her select this time stamp to sort all static attributes in the heatmap by this rather late time point of the temporal data. The list panel on the right (*SUP-Fig6*) guides her to the *Physicians per 1000* attribute, which shows the strongest (positive) correlation to the reference attribute. She finds that, during the phase of the steep increase, the number of physicians per 1000 was decreasing in its positive correlation, indicating that there may have been stress in the medical system, struggling to cope with the situation. In turn, the effect of a plateau in confirmed cases right after may have its cause by the positive correlation of physicians per 1000, indicating a return to normality. She identifies the same phenomena for the *Health expenditure* attribute, which may be correlated also to the physicians per 1000 attribute. After an in-depth analysis of another time point in (*SUP-Fig7*), Christina transitions to the correlation analysis ($\mathbf{T}_{\mathrm{Corr}}$) with time series attributes (*SUP-Fig8*). Here, she identifies much more heterogeneity in the correlation heatmap, which can be explained by the fact that the correlation measure now has always two time series attributes as input, as opposed to one static input attribute before. By using the list-based guidance view on the right she identifies a strong positive correlation of the *Debt relief* attribute with the reference attribute (cumulative conformed). By looking into the heatmap, she finds an increase in the positive correlation of the debt relief. She infers that, when cases are increasing, governments offer debt relief. In contrast, face coverings show negative correlations with cumulative cases over time, as people may be scared to take their masks off. She continues her analysis by sorting another time point, where the cumulative cases have reached a plateau (*SUP-Fig9*). She identifies the strong correlation of the *School closing* attribute. She raises the hypothesis that closed schools helped to keep the numbers low. Finally, Christina inspects the detailed view of one single correlation attribute (*Stringency index*) with the reference attribute (cumulative confirmed), for a single time stamp. A popup with the two temporal attributes and the correlation scatter plot appears (*SUP-Fig10*). She clearly identifies that, when the cumulative cases plateaued, the Stringency index representing its nine composite measures went down before, identified through a very strong positive correlation.

7.2. Case Study: Stocks Data

In this case study, we collaborate with Daniel, an economics researcher from the Faculty of Business, Economics, and Informatics at the University of Zurich, working with stock data. Daniel first familiarizes himself with the temporal attributes provided with the case, and without further ado starts focusing on the *Prices* attribute (*SUP-Fig11*). As he deems the absolute values of the stock prices sub-optimal for downstream analyses, he decides to apply a normalization routine to the attribute. He opens the derive attribute popup and creates the *Price [derived]* attribute, including two operations on absolute prices (*SUP-Fig12*). First, he adds a value shift operation for time point 0, so that each value starts at 0 at the first time point in Q1 of 2010. He also adds a normalization routine to transform the prices into a relative value domain. The *Prices [derived]* attribute likewise serves as his similarity configuration ($\mathbf{T}_{\text{Simi}}$). Daniel continues with filtering for sectors of the companies, to have a more homogeneous peer set (*SUP-Fig13*). Here, he decides for stocks *Consumer Cyclical* of companies. As a next step, Daniel switches to an object-oriented perspective and selects *Macy's Inc.* as his preferred query object ($\mathbf{T}_{\text{Query}}$), representative of consumer cyclical stocks (*SUP-Fig14*, combo box top right). He proceeds with adding and analyzing the *Asset Turnover* attribute in the Similarity Configuration Table (*SUP-Fig14*), to double-check for the seasonal periodicity pattern between the four quarters of every year, to be expected for consumer cyclical stocks. Daniel decides to not apply any geographical filter, to maintain the maximum geographical variety. With that, he feels prepared for the exploration phase, using the identified subset of stocks as his peer group ($\mathbf{T}_{\text{Peers}}$).

In the exploration phase, Daniel inspects and refines his peer group again, now consisting of 29 stocks (*SUP-Fig15*). He proceeds with determining the *Price [derived]* attribute as his reference attribute and decides to explore correlations in temporal attributes ($\mathbf{T}_{\text{Corr}}$). Daniel's first observations during exploration are the many positive correlations of attributes to the price attribute. Here, explaining identified relations is of Daniel's particular interest, as the identification of attributes that have an impact on price effects is one of the most interesting activities in his work life as an economist. Also, two mostly negatively correlated attributes exist: the *Asset Turnover* and the *Effective Tax Rate*. Further, Daniel identifies many attributes with very similar temporal correlation patterns, most likely due to a high correlation: *Debt Equity Ratio*, *Debt Ratio*, *Debt to Assets*, and *Debt to Equity*. As a next step, Daniel focuses on individual time steps, such as the one in the Covid-19 pandemic in 2020 where, surprisingly, many stock prices increased again (*SUP-Fig16*). The list-based guidance panel reveals that the *Price To Sales Ratio* is the most (positively) correlated attribute of all, which makes sense as this market coefficient is directly computed from the stock price. Other attributes with high correlation include the *Asset Turnover*, and the *Market Capitalization*. Daniel identifies a similar situation for a different time point (*SUP-Fig17*). Sorting attributes by this time point reveals a nice separation between positively and negatively correlated attributes. He utilizes this ordering to identify an attribute of interest and starts a detailed correlation analysis between the *Price [derived]* and the *Price To Sales Ratio* (*SUP-Fig18*). He uses the slider to browse through different time points and analyzes the correlation details in the scatter plot.

7.3. Early Feedback from Experts

For the case studies, both experts provided early feedback on their experience with LFPeers. While most feedback was positive, there were also suggestions for improvement.

One of the positive aspects both experts expressed was the flexibility of exploring multivariate data. Specifically, the experts appreciated the ability to transform attributes, such as performing normalizations of data attributes at run-time, and to create derived attributes as needed. Also, the ability to visualize the time series of multiple attributes in the Similarity Configuration Table enabled them to compare and select attributes of interest. They also valued the ability to select a time series attribute to find similar peers through simple and intuitive interaction. In specific, Christina was keen on finding attributes of interest and on comparing time series, specifically, the vaccination policy, or school closing. In turn, Daniel spent much time analyzing correlations in the exploration phase and had interesting findings, especially on correlations over time. The bipolar visual encoding of positive and negative correlation was useful for him to assess the volatility of attributes. Using these findings, he was keen on investigating how these variables influenced the peer set during the time period he selected. Similarly, Christina was able to use the correlation view to derive valuable insights about relations in the data.

However, an aspect of criticism expressed by both experts was on the intuitiveness of LFPeers. For example, Christina would have preferred to use a button rather than a drag-and-drop interaction, provided to add attributes to the Similarity Configuration Table. In turn, Daniel was concerned about how to find out the most meaningful number of peers before entering the exploration phase, despite being aware of the supporting views.

8. Discussion

Interactive Similarity Search. LFPeers provides means to search for objects (peers) with up to four different user-defined

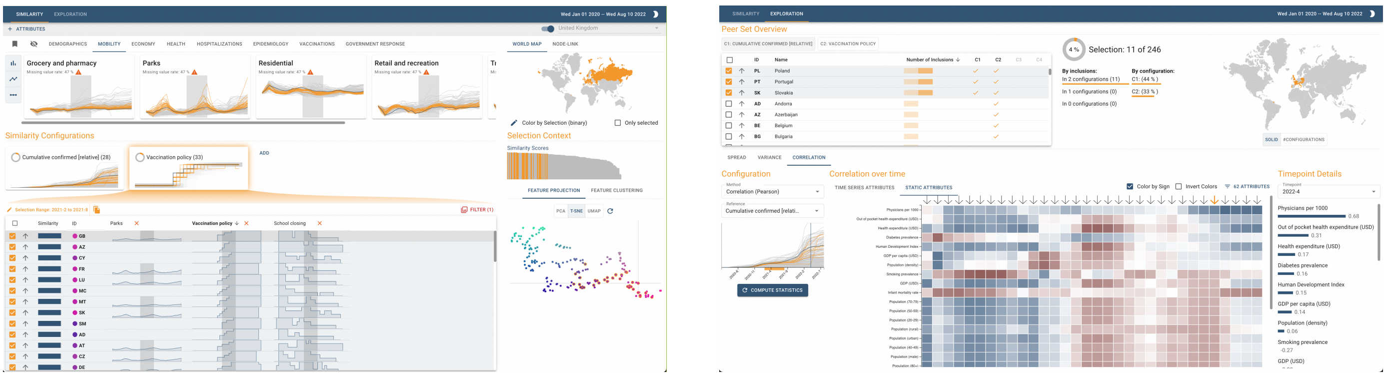

Fig. 10. The Covid-19 case at a glance. Search interface (left): An early phase in the analysis process as described in the Covid-19 scenario 7.1. The expert has drawn relevant attributes into the bookmark group, including the derived *Cumulative confirmed [relative]* time series. Exploration interface (right): After further search steps, the expert has selected a final peer group of 11 countries, and inspects correlated attributes to the *Cumulative confirmed [relative]* time series.

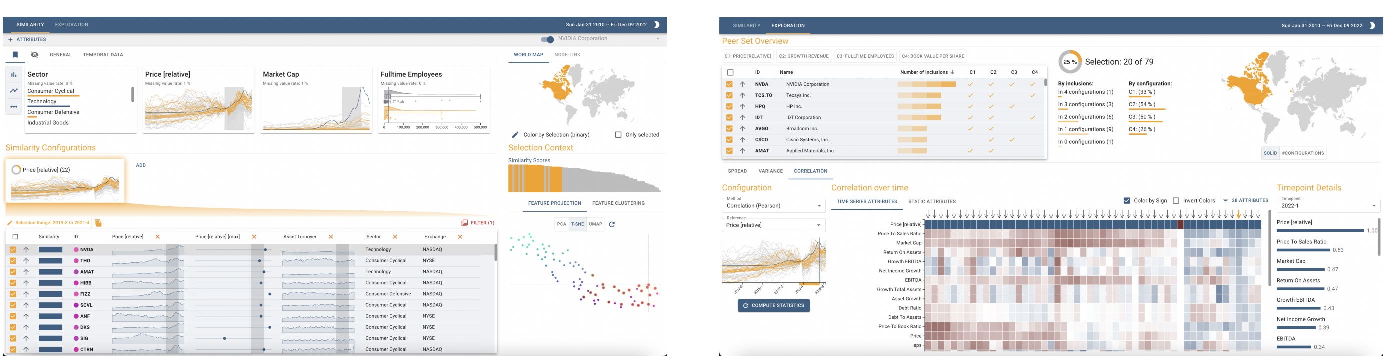

Fig. 11. The stocks case at a glance. Search interface (left): An early phase in the analysis process as described in the Stocks scenario 7.2. The expert has drawn relevant attributes into the bookmark group, including the derived *Price [relative]* time series, that is aligned to 0 at the start of the pandemic era. The selection of peers shall focus on companies that have shown upward trends in this time, so the expert has used a simple difference distance metric between the start and end value of the time range to determine the selection.

similarity configurations in parallel, which was found interesting and powerful by both domain experts. Despite our proposed solution, we identified that implementations of concepts for interactive similarity search in general may benefit from reported best practices, as well as from guidelines in the design of interactive similarity search interfaces. One of the central research aspects would include the coupling of the interactive interface with the specifics of the underlying data and involved data transformations and processes, yielding an interesting design space.

Computational Scalability. We have decided for an online process on data transformation and feature extraction, to enable users to define time series similarity at runtime. While for users, this offers a high degree of flexibility and control of the process, this type of visual analytics support raises interesting challenges from a computational perspective. Two angles of attack include questions on the possibility of pre-computation of individual process modules, as well as the parallelization of operations, both to be more scalable in an online process. With the algorithmic capabilities and implementation described in Section6, LFPeers can execute similarity search operations for 1,000 objects at interactive speed. We assume short delays for 10,000 objects which may still be acceptable, as the computation could be done asynchronously in the background. However, for larger datasets, the computation, memory allocation, JavaScript support, and the visual scalability may run into problems.

Temporal Flexibility. LFPeers is designed for situations where an endpoint t_0 exists that can be used to clearly separate the earlier search for peers and the later exploration of peer behavior, since this is, e.g., often the case with endpoints in healthcare or situations with concrete events. An extension of LFPeers could include a time interval (absolute time) or a duration (relative time) instead of a discrete endpoint, to allow for learning-from-peers scenarios that require more temporal flexibility. Similarly, it would be interesting to study a more flexible implementation of the exploration endpoint t_1. An inspiring comment in the case study along similar lines was to allow warping concepts in the correlation computation, for the identification of effects with temporal offsets and time-lagged attributes.

Closing the Loop. During the case studies, we observed a curiosity of both experts in the exploration phase for sorting attributes by correlations and studying individual peer objects according to their differences from the previously set query object. This knowledge gained during the exploration phase was useful to explain deviations and to learn from peers. Also, this gained knowledge was used as a feedback loop, to start a new search phase after a successful learning-from-peers iteration. Extending the conceptual workflow and its five consecutive tasks to a cyclic structure ($\mathbf{T}_5$ re-informs $\mathbf{T}_1$) would be an interesting conceptual direction for future work.

Usability. Despite the iterative design and involvement of experts in case studies, LFPeers has yet not been used to conduct a user study to assess the usability aspects of the visual analytics interfaces. Two concrete evaluation targets would be to quantitatively assess the usability of LFPeers (e.g., by applying the System Usability Scale (SUS) [88]), and to identify usability barriers locally in the workflow, eventually caused by individual interfaces. A type of observational study would be to protocol the order of tasks conducted by users ($\mathbf{T}_{\text{Query}}$, $\mathbf{T}_{\text{Simi}}$, $\mathbf{T}_{\text{Peers}}$, $\mathbf{T}_{\text{Var}}$, $\mathbf{T}_{\text{Corr}}$) to study the iterative nature of the workflow and to identify task subsequences deviating from the general workflow. As a side effect, it would also be interesting to study users further, e.g., to identify different types of persona and, successively, make informed decisions on the tool refinement and specialization for future use cases and identify guidelines for the design of interfaces for the learning-from-peers concept in general.

9. Conclusion

We have presented a visual analytics approach for the identification, analysis, and explanation of relations in objects with time-oriented data attributes, to enable users to learn from peers. At a glance, our approach supports users in the search for a group of similar peer objects in an early phase of the time series, as well as in the exploration of variations within this group as occurring in later phases of the temporal data. In addition, users have the means to explain these variations using correlation analysis and to learn from the peer group. We have shown that our approach, originally motivated by Covid-19 data and use cases, can be generalized to other domains, and we believe that it can be extended to a large variety of other cases as well. Searching for time series by similarity comes with many degrees of freedom that pose challenges. In many real-world scenarios, the similarity is also often determined by more than just a single (temporal) attribute, and data objects are semantically linked, grouped, or related in various ways to others. We have shown that allowing users to determine similarity configurations interactively is a key enabling factor to extend this approach to other domains.

References

[1] Aigner, W, Miksch, S, Schumann, H, Tominski, C. Visualization of Time-Oriented Data. Human-Computer Interaction Series; Springer; 2011. ISBN 978-0-85729-078-6. doi:10.1007/978-0-85729-079-3.

[2] Kehrer, J, Hauser, H. Visualization and visual analysis of multifaceted scientific data: A survey. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2013;19(3):495–513. doi:10.1109/TVCG.2012.110.

[3] Afzal, S, Ghani, S, Jenkins-Smith, HC, Ebert, DS, Hadwiger, M, Hoteit, I. A visual analytics based decision making environment for covid-19 modeling and visualization. In: IEEE Visualization Conference (VIS). 2020, p. 86–90. doi:10.1109/VIS47514.2020.00024.

[4] Polonsky, JA, Baidjoe, A, Kamvar, ZN, Cori, A, Durski, K, Edmunds, WJ, et al. Outbreak analytics: a developing data science for informing the response to emerging pathogens. Philosophical Transactions of the Royal Society B: Biological Sciences 2019;374(1776):20180276. doi:10.1098/rstb.2018.0276.

[5] Yañez, A, Duggan, J, Hayes, C, Jilani, M, Connolly, M. Pandemcap: Decision support tool for epidemic management. In: IEEE Workshop on Visual Analytics in Healthcare (VAHC). 2017, p. 24–30. doi:10.1109/VAHC.2017.8387497.

[6] Abdelhamid, SE, Kuhlman, CJ, Marathe, MV, Ravi, SS. Interactive exploration and understanding of contagion dynamics in networked populations. In: Conference on Behavioral, Economic and Socio-cultural Computing (BESC). 2016, p. 1–6. doi:10.1109/BESC.2016.7804480.

[7] Leite, RA, Schetinger, V, Ceneda, D, Henz, B, Miksch, S. COVIs: Supporting temporal visual analysis of covid-19 events usable in data-driven journalism. In: IEEE Visualization Conference (VIS). 2020, p. 56–60. doi:10.1109/VIS47514.2020.00018.

[8] Wongsuphasawat, K, Guerra Gómez, JA, Plaisant, C, Wang, TD, Taieb-Maimon, M, Shneiderman, B. Lifeflow: Visualizing an overview of event sequences. In: Conference on Human Factors in Computing Systems (CHI). ACM. ISBN 9781450302289; 2011, p. 1747–1756. doi:10.1145/1978942.1979196.

[9] Bach, B, Shi, C, Heulot, N, Madhyastha, T, Grabowski, T, Dragicevic, P. Time curves: Folding time to visualize patterns of temporal evolution in data. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2016;22(1):559–568. doi:10.1109/TVCG.2015.2467851.

[10] Ziegler, H, Jenny, M, Gruse, T, Keim, DA. Visual market sector analysis for financial time series data. In: IEEE Conference on Visual Analytics Science and Technology (VAST). IEEE Computer Society; 2010, p. 83–90. doi:10.1109/VAST.2010.5652530.

[11] Cibulski, L, May, T, Preim, B, Bernard, J, Kohlhammer, J. Visualizing time series consistency for feature selection. In: Journal of WSCG. 2019, p. 93–102. doi:10.24132/JWSCG.2019.27.2.2.

[12] Hu, Y, Wu, S, Xia, S, Fu, J, Chen, W. Motion track: Visualizing variations of human motion data. In: IEEE Pacific Visualization Symposium PacificVis 2010, Taipei, Taiwan, March 2-5, 2010. IEEE Computer Society; 2010, p. 153–160. doi:10.1109/PACIFICVIS.2010.5429596.

[13] Li, J, Chen, S, Andrienko, GL, Andrienko, NV. Visual exploration of spatial and temporal variations of tweet topic popularity. In: EuroVis Workshop on Visual Analytics (EuroVA). Eurographics Association; 2018, p. 7–11. doi:10.2312/eurova.20181105.

[14] May, T. Working with patterns in large multivariate datasets - Karnaugh-Veitch-maps revisited. In: International Conference on Information Visualisation IV. IEEE Computer Society; 2007, p. 277–285. doi:10.1109/IV.2007.145.

[15] Vesanto, J, Ahola, J. Hunting for correlations in data using the self-organizing map. In: International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99), Rochester, New York, USA, June 22-25, 1999. ICSC Academic Press; 1999, p. 279–285.

[16] Eirich, J, Bonart, J, Jäckle, D, Sedlmair, M, Schmid, U, Fischbach, K, et al. IRVINE: A design study on analyzing correlation patterns of electrical engines. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2022;28(1):11–21. doi:10.1109/TVCG.2021.3114797.

[17] Burmeister, J, Bernard, J, Kohlhammer, J. LFPeers: Temporal Similarity Search in Covid-19 Data. In: EuroVis Workshop on Visual Analytics (EuroVA). The Eurographics Association. ISBN 978-3-03868-150-2; 2021,doi:10.2312/eurova.20211098.

[18] Wahltinez, O, et al. COVID-19 Open-Data: curating a fine-grained, global-scale data repository for SARS-CoV-2. Work in progress 2020;URL: https://goo.gle/covid-19-open-data.

[19] Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. In: IEEE Symposium on Visual Languages. IEEE Computer Society; 1996, p. 336–343. doi:10.1109/VL.1996.545307.

[20] Keim, DA, Andrienko, GL, Fekete, J, Görg, C, Kohlhammer, J, Melançon, G. Visual analytics: Definition, process, and challenges. In: Information Visualization – Human-Centered Issues and Perspectives; vol. 4950 of *Lecture Notes in Computer Science*. Springer; 2008, p. 154–175. URL: https://doi.org/10.1007/978-3-540-70956-5_7. doi:10.1007/978-3-540-70956-5_7.

[21] Keim, DA, Kohlhammer, J, Ellis, GP, Mansmann, F. Mastering the Information Age - Solving Problems with Visual Analytics. Eurographics Association; 2010. URL: http://diglib.eg.org/EG/Publications/bookstore/Data/pe_vismaster2010.htm.

[22] Sacha, D, Stoffel, A, Stoffel, F, Kwon, BC, Ellis, GP, Keim, DA. Knowledge generation model for visual analytics. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2014;20(12):1604–1613. doi:10.1109/TVCG.2014.2346481.

[23] Du, F, Plaisant, C, Spring, N, Shneiderman, B. Finding similar people to guide life choices: Challenge, design, and evaluation. In: Conference on Human Factors in Computing Systems (CHI). 2017, p. 5498–5544. doi:10.1145/3025453.3025777.

[24] Du, F, Plaisant, C, Spring, N, Crowley, K, Shneiderman, B. Eventaction: A visual analytics approach to explainable recommendation for event sequences. ACM Transactions on Interactive Intelligent Systems (TiiS) 2019;9(4):1–31. doi:10.1145/3301402.

[25] Marchionini, G, Shneiderman, B. Finding facts vs. browsing knowledge in hypertext systems. Computer 1988;21(1):70–80. doi:10.1109/2.222119.

[26] Wildemuth, BM, O'Neill, AL. The "known" in known-item searches: Empirical support for user-centered design (research note). College & Research Libraries 1995;56(3):265–281.

[27] Bates, MJ. The design of browsing and berrypicking techniques for the online search interface. Online review 1989;13(5):407–424. doi:10.1108/eb024320.

[28] Ahlberg, C, Shneiderman, B. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In: Conference on Human Factors in Computing Systems (CHI). ACM; 1994, p. 313–317. doi:10.1145/191666.191775.

[29] Tunkelang, D. Faceted Search. Synthesis Lectures on Information Concepts, Retrieval, and Services; Morgan & Claypool Publishers; 2009. doi:10.2200/S00190ED1V01Y200904ICR005.

[30] Andrienko, NV, Andrienko, GL. Exploratory analysis of spatial and temporal data - a systematic approach. Springer; 2006. ISBN 978-3-540-25994-7. doi:10.1007/3-540-31190-4.

[31] Marchionini, G. Exploratory search: from finding to understanding. Commun ACM 2006;49(4):41–46. doi:10.1145/1121949.1121979.

[32] White, RW, Roth, RA. Exploratory search: Beyond the query-response paradigm. Synthesis lectures on information concepts, retrieval, and services 2009;1(1):1–98.

[33] Borgman, CL, Wallis, JC, Enyedy, N. Building digital libraries for scientific data: An exploratory study of data practices in habitat ecology. In: Research and Advanced Technology for Digital Libraries ECDL; vol. 4172 of *Lecture Notes in Computer Science*. Springer; 2006, p. 170–183. doi:10.1007/11863878_15.

[34] Hey, T, Tansley, S, Tolle, KM, editors. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research; 2009. ISBN 978-0982544204. URL: http://research.microsoft.com/en-us/collaboration/fourthparadigm/.

[35] Ahlberg, C, Shneiderman, B. Visual information seeking using the filmfinder. In: Conference on Human Factors in Computing Systems (CHI). ACM; 1994, p. 433–434. doi:10.1145/259963.260431.

[36] Herrmannova, D, Knoth, P. Visual search for supporting content exploration in large document collections. D-Lib Magazine 2012;18(7/8). doi:10.1045/july2012-herrmannova.

[37] Oren, E, Delbru, R, Decker, S. Extending faceted navigation for RDF data. In: The Semantic Web (ISWC); vol. 4273 of *Lecture Notes in Computer Science*. Springer; 2006, p. 559–572. doi:10.1007/11926078_40.

[38] Dörk, M, Carpendale, S, Collins, C, Williamson, C. Visgets: Coordinated visualizations for web-based information exploration and discovery. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2008;14(6):1205–1212. doi:10.1109/TVCG.2008.175.

[39] Dörk, M, Riche, NH, Ramos, GA, Dumais, ST. Pivotpaths: Strolling through faceted information spaces. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2012;18(12):2709–2718. doi:10.1109/TVCG.2012.252.

[40] Heimerl, F, Lohmann, S, Lange, S, Ertl, T. Word cloud explorer: Text analytics based on word clouds. In: Hawaii International Conference on System Sciences, HICSS. IEEE Computer Society; 2014, p. 1833–1842. doi:10.1109/HICSS.2014.231.

[41] van Ham, F, Perer, A. "search, show context, expand on demand": Supporting large graph exploration with degree-of-interest. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2009;15(6):953–960. doi:10.1109/TVCG.2009.108.

[42] Krause, J, Perer, A, Stavropoulos, H. Supporting iterative cohort construction with visual temporal queries. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2016;22(1):91–100. doi:10.1109/TVCG.2015.2467622.

[43] Hearst, MA. Search User Interfaces. 1st ed.; Cambridge University Press; 2009. ISBN 0521113792.

[44] Hochheiser, H, Shneiderman, B. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. Information Visualization 2004;3(1):1–18.

[45] Bernard, J, Daberkow, D, Fellner, DW, Fischer, K, Koepler, O, Kohlhammer, J, et al. VisInfo: a digital library system for time series research data based on exploratory search - a user-centered design ap-

proach. International Journal on Digital Libraries (IJoDL) 2015;16(1):37–59. doi:10.1007/s00799-014-0134-y.

[46] Bernard, J, Wilhelm, N, Krüger, B, May, T, Schreck, T, Kohlhammer, J. MotionExplorer: Exploratory search in human motion capture data based on hierarchical aggregation. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2013;19(12):2257–2266. doi:10.1109/TVCG.2013.178.

[47] Holz, C, Feiner, S. Relaxed selection techniques for querying time-series graphs. In: ACM symposium on User interface software and technology. 2009, p. 213–222.

[48] Ryall, K, Lesh, N, Lanning, T, Leigh, D, Miyashita, H, Makino, S. Querylines: approximate query for visual browsing. In: Conference on Human Factors in Computing Systems (CHI). ACM; 2005, p. 1765–1768. doi:10.1145/1056808.1057017.

[49] Correll, M, Gleicher, M. The semantics of sketch: Flexibility in visual query systems for time series data. In: Conference on Visual Analytics Science and Technology (VAST). IEEE Computer Society; 2016, p. 131–140. doi:10.1109/VAST.2016.7883519.

[50] Gogolou, A, Tsandilas, T, Palpanas, T, Bezerianos, A. Comparing similarity perception in time series visualizations. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2018;25(1):523–533.

[51] Kandel, S, Heer, J, Plaisant, C, Kennedy, J, van Ham, F, Riche, NH, et al. Research directions in data wrangling: Visualizations and transformations for usable and credible data. Inf Vis 2011;10(4):271–288. doi:10.1177/1473871611415994.

[52] Bernard, J, Ruppert, T, Goroll, O, May, T, Kohlhammer, J. Visual-interactive preprocessing of time series data. In: Proceedings of SIGRAD 2012, Interactive Visual Analysis of Data, Växjö, Sweden, November 29-30, 2012; vol. 81 of *Linköping Electronic Conference Proceedings*. Linköping University Electronic Press; 2012, p. 39–48. URL: http://www.ep.liu.se/ecp/article.asp?issue=081&article=006&volume=.

[53] Arbesser, C, Spechtenhauser, F, Mühlbacher, T, Piringer, H. Visplause: Visual data quality assessment of many time series using plausibility checks. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2017;23(1):641–650. doi:10.1109/TVCG.2016.2598592.

[54] Bernard, J, Hutter, M, Reinemuth, H, Pfeifer, H, Bors, C, Kohlhammer, J. Visual-interactive preprocessing of multivariate time series data. Computer Graphics Forum (CGF) 2019;38(3):401–412. doi:10.1111/cgf.13698.

[55] Bernard, J, Ruppert, T, Scherer, M, Schreck, T, Kohlhammer, J. Guided discovery of interesting relationships between time series clusters and metadata properties. In: International Conference on Knowledge Management and Knowledge Technologies I-KNOW. ACM; 2012, p. 22. doi:10.1145/2362456.2362485.

[56] Fu, T. A review on time series data mining. Eng Appl Artif Intell 2011;24(1):164–181. doi:10.1016/j.engappai.2010.09.007.

[57] Keogh, EJ, Kasetty, S. On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Min Knowl Discov 2003;7(4):349–371. doi:10.1023/A:1024988512476.

[58] Ding, H, Trajcevski, G, Scheuermann, P, Wang, X, Keogh, EJ. Querying and mining of time series data: experimental comparison of representations and distance measures. Proc VLDB Endow 2008;1(2):1542–1552. doi:10.14778/1454159.1454226.

[59] Zhao, J, Chevalier, F, Pietriga, E, Balakrishnan, R. Exploratory analysis of time-series with chronolenses. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2011;17(12):2422–2431.

[60] Zhao, J, Chevalier, F, Balakrishnan, R. Kronominer: using multi-foci navigation for the visual exploration of time-series data. In: Conference on Human Factors in Computing Systems (CHI). 2011, p. 1737–1746.

[61] Sacha, D, Zhang, L, Sedlmair, M, Lee, JA, Peltonen, J, Weiskopf, D, et al. Visual interaction with dimensionality reduction: A structured literature analysis. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2017;23(1):241–250. doi:10.1109/TVCG.2016.2598495.

[62] Ali, M, Alqahtani, A, Jones, MW, Xie, X. Clustering and classification for time series data in visual analytics: A survey. IEEE Access 2019;7:181314–181338. doi:10.1109/ACCESS.2019.2958551.

[63] McLachlan, P, Munzner, T, Koutsofios, E, North, SC. Liverac: interactive visual exploration of system management time-series data. In: Conference on Human Factors in Computing Systems (CHI). ACM; 2008, p. 1483–1492. doi:10.1145/1357054.1357286.

[64] van Wijk, JJ, van Selow, ER. Cluster and calendar based visualization of time series data. In: IEEE Symposium on Information Visualization 1999 (INFOVIS'99), San Francisco, California, USA, October 24-29, 1999. IEEE Computer Society; 1999, p. 4–9. doi:10.1109/INFVIS.1999.801851.

[65] Schreck, T, Bernard, J, von Landesberger, T, Kohlhammer, J. Visual cluster analysis of trajectory data with interactive kohonen maps. Inf Vis 2009;8(1):14–29. doi:10.1057/ivs.2008.29.

[66] Sacha, D, Kraus, M, Bernard, J, Behrisch, M, Schreck, T, Asano, Y, et al. Somflow: Guided exploratory cluster analysis with self-organizing maps and analytic provenance. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2018;24(1):120–130. doi:10.1109/TVCG.2017.2744805.

[67] Wilhelm, N, Vögele, A, Zsoldos, R, Licka, T, Krüger, B, Bernard, J. Furyexplorer: visual-interactive exploration of horse motion capture data. In: Visualization and Data Analysis 2015, San Francisco, CA, USA, February 9-11, 2015; vol. 9397 of *SPIE Proceedings*. SPIE; 2015, p. 93970F. doi:10.1117/12.2080001.

[68] Bernard, J, Steiger, M, Widmer, S, Lücke-Tieke, H, May, T, Kohlhammer, J. Visual-interactive exploration of interesting multivariate relations in mixed research data sets. Computer Graphics Forum (CGF) 2014;33(3):291–300. doi:10.1111/cgf.12385.

[69] Koldijk, S, Bernard, J, Ruppert, T, Kohlhammer, J, Neerincx, MA, Kraaij, W. Visual analytics of work behavior data - insights on individual differences. In: Eurographics Conference on Visualization, EuroVis 2015 (Short Papers). Eurographics Association; 2015, p. 79–83. doi:10.2312/eurovisshort.20151129.

[70] Wang, TD, Plaisant, C, Quinn, AJ, Stanchak, R, Murphy, S, Shneiderman, B. Aligning temporal data by sentinel events: discovering patterns in electronic health records. In: Conference on Human Factors in Computing Systems (CHI). 2008, p. 457–466.

[71] Bernard, J, Wilhelm, N, Scherer, M, May, T, Schreck, T. TimeSeriesPaths: Projection-Based Explorative Analysis of Multivariate Time Series Data. Journal of WSCG 2012;20(2):97–106. URL: http://wscg.zcu.cz/wscg2012/Index.htm.

[72] Ziegler, H, Nietzschmann, T, Keim, DA. Visual exploration and discovery of atypical behavior in financial time series data using two-dimensional colormaps. In: International Conference on Information Visualisation IV. IEEE Computer Society; 2007, p. 308–315. doi:10.1109/IV.2007.124.

[73] Stoffel, F, Fischer, F, Keim, DA. Finding anomalies in time-series using visual correlation for interactive root cause analysis. In: Workshop on Visualization for Cyber Security. 2013, p. 65–72.

[74] Bernard, J, Steiger, M, Widmer, S, Lücke-Tieke, H, May, T, Kohlhammer, J. Visual-interactive Exploration of Interesting Multivariate Relations in Mixed Research Data Sets. Computer Graphics Forum (CGF) 2014;33(3):291–300. doi:10.1111/cgf.12385.

[75] Buono, P, Aris, A, Plaisant, C, Khella, A, Shneiderman, B. Interactive pattern search in time series. In: Visualization and Data Analysis (VDA); vol. 5669. SPIE; 2005, p. 175–186. doi:10.1117/12.587537.

[76] Keogh, EJ, Chakrabarti, K, Pazzani, MJ, Mehrotra, S. Dimensionality reduction for fast similarity search in large time series databases. Knowl Inf Syst 2001;3(3):263–286. doi:10.1007/PL00011669.

[77] Bernard, J. Exploratory search in time-oriented primary data. Dissertation, PhD Thesis (Dissertation) 2015;URL: http://tuprints.ulb.tu-darmstadt.de/5173/.

[78] Salvador, S, Chan, P. Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis 2007;11(5):561–580.

[79] Espadoto, M, Martins, RM, Kerren, A, Hirata, NS, Telea, AC. Towards a quantitative survey of dimension reduction techniques. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2019;.

[80] Van der Maaten, L, Hinton, G. Visualizing data using t-sne. Journal of machine learning research 2008;9(11).

[81] McInnes, L, Healy, J. UMAP: uniform manifold approximation and projection for dimension reduction. CoRR 2018;abs/1802.03426. URL: http://arxiv.org/abs/1802.03426. arXiv:1802.03426.

[82] Bernard, J, Steiger, M, Mittelstädt, S, Thum, S, Keim, D, Kohlhammer, J. A survey and task-based quality assessment of static 2d colormaps. vol. 9397. SPIE Press; 2015, p. 93970M–93970M–16. doi:10.1117/12.2079841.

[83] Vesanto, J. SOM-based data visualization methods. Intelligent data analysis 1999;3(2):111–126.

[84] Javed, W, McDonnel, B, Elmqvist, N. Graphical perception of multiple

time series. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2010;16(6):927–934. doi:10.1109/TVCG.2010.162.

[85] Bernard, J, Hutter, M, Reinemuth, H, Pfeifer, H, Bors, C, Kohlhammer, J. Visual-interactive preprocessing of multivariate time series data. Computer Graphics Forum (CGF) 2019;38(3):401–412. doi:10.1111/cgf.13698.

[86] Gleicher, M, Albers, D, Walker, R, Jusufi, I, Hansen, CD, Roberts, JC. Visual comparison for information visualization. Inf Vis 2011;10(4):289–309. doi:10.1177/1473871611416549.

[87] Allen, M, Poggiali, D, Whitaker, K, Marshall, TR, Kievit, RA. Raincloud plots: a multi-platform tool for robust data visualization. Wellcome open research 2019;4. doi:10.7287/peerj.preprints.27137v1.

[88] Brooke, J, et al. Sus-a quick and dirty usability scale. Usability evaluation in industry 1996;189(194):4–7.