DOI: 10.1111/j.1467-8659.2011.01920.x
Eurographics / IEEE Symposium on Visualization 2011 (EuroVis 2011)
H. Hauser, H. Pfister, and J. J. van Wijk
(Guest Editors)
Volume 30 (2011), Number 3

Interactive Visual Analysis of Temporal Cluster Structures

C. Turkay[1], J. Parulek[1], N. Reuter[2], and H. Hauser[1]

[1]Department of Informatics, University of Bergen [2]BCCS, University of Bergen

Abstract

Cluster analysis is a useful method which reveals underlying structures and relations of items after grouping them into clusters. In the case of temporal data, clusters are defined over time intervals where they usually exhibit structural changes. Conventional cluster analysis does not provide sufficient methods to analyze these structural changes, which are, however, crucial in the interpretation and evaluation of temporal clusters. In this paper, we present two novel and interactive visualization techniques that enable users to explore and interpret the structural changes of temporal clusters. We introduce the temporal cluster view, which visualizes the structural quality of a number of temporal clusters, and temporal signatures, which represents the structure of clusters over time. We discuss how these views are utilized to understand the temporal evolution of clusters. We evaluate the proposed techniques in the cluster analysis of mixed lipid bilayers.

Categories and Subject Descriptors (according to ACM CCS): Computing Methodologies [I.3.m]: Computer Graphics—Miscellaneous

1. Introduction

With the advance of data acquisition and simulation systems, large amounts of data with a high number of dimensions and temporally varying values are produced. In various fields like bioinformatics, financial analysis and engineering, it is of great importance to explore and understand the groups of data which share common characteristics over time. These groups are usually analyzed further to gain insight into the processes that are governed by these common characteristics. Cluster analysis is a widely used method to discover grouping structures in both static and time-varying data. This analysis results in a set of clusters, each of which represents a group of similar items with respect to certain features of the data. However, when performing cluster analysis on temporal datasets, interpreting and evaluating the resulting clusters is not as straightforward as it is with static data.

Most of the algorithms developed for clustering time series (temporal) data are either modifications of the static data clustering algorithms, or time-series are converted into static representations such that existing algorithms can be used [Lia05]. Therefore, these clustering algorithms focus mainly on the design of a proper distance function to use in clustering or in the conversion of the data into feature vectors of lower dimensionality. These custom distance functions and conversion operations applied to large, high-dimensional time series may easily produce low-quality clusters [WSH06]. As a consequence, the interpretation and evaluation of clusters become a very important part of cluster analysis. Current methods for cluster assessment, however, are mainly tailored for static data [Lia05], yielding a need for new mechanisms to analyze temporal clusters.

In the following, we illustrate a simple situation where advanced analysis techniques are required to understand the variation of time-dependent cluster structures. We consider a simple scenario as illustrated in Fig. 1. In this setting, two well separated and equally sized groups merge into a single, heterogeneous group at time t_1 and split into two groups again at time t_2. This simple scenario demonstrates a typical example of structural changes which clusters can exhibit over time. Also note that, clustering different time intervals (i.e., t_0, t_1 or t_2) yields completely different clusters.

As the overall clustering structure changes temporally in time-series data, cluster analysis of such data is generally performed over intervals of time [SS05]. Therefore, unlike clusters of static data, temporal clusters have temporal spans in addition to the group of items they represent. Due to the fact that temporal clusters do not exhibit stable structures usually, both cluster-cluster relations and the structure of temporal clusters vary over time. However, if an experienced user could evaluate such variations, then she/he could conse-

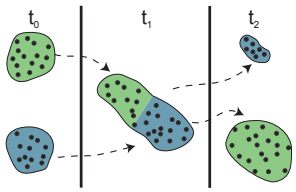

Figure 1: *An example of structural changes in clusters of temporal data. Two well-separated clusters (at t_0) merge into a single group at t_1 and split into two groups again at t_2.*

quently discard or update the clusters. The analysis of these variations are not really addressed by the current methods and techniques in cluster analysis. In order to interpret and evaluate temporal clusters, the analyst has to answer at least two questions; firstly, "How does the quality of clusters vary over time?" and secondly, "What type of structural changes do clusters exhibit?".

In this paper, we propose two novel and interactive visualization techniques to analyze temporal clusters. We firstly introduce the *temporal cluster view* that visualizes the structural quality of temporal cluster sets over time. Secondly, we present *temporal signatures* which are visual summaries of temporal cluster structures. The cluster view provides mechanisms to visualize and interactively analyze a set of temporal clusters that are computed from different time intervals. This view also encodes *silhouette coefficients* [Rou87], which are quite widely used cluster structure metrics. They are used to evaluate the structural quality of cluster sets. Temporal signatures are representations of statistical properties of clusters over time. These properties are based on *cluster cohesion* which represents the tightness of its items, and *cluster homogeneity* which correspond to the uniformity of the distribution of the member items [TSK06].

When used in conjunction, these two views provide intuitive mechanisms to analyze and evaluate temporal clusters. They are utilized to explore structural changes in clusters; namely, splitting, merging, and changes in cluster size. We present these two views in an interactive visual analysis framework. To summarize, our contributions in this paper are:

- The temporal cluster view, visualizing a number of temporal clusters together with their structural quality variation.
- Temporal signatures, that are visual representations of the structural changes of groups over time.
- Interactive visual analysis procedures for temporal cluster analysis with the help of these two views.

2. Related Work

Our work relates to the analysis of temporal clusters using interactive techniques and visual representations for temporally varying structures. Thus, the related literature is presented in three subsections:

Analyzing clusters – Vectorized radial visualizations are used in exploring different clustering results by projecting data records on a vectorized cluster space [SGM08]. This approach proves to be useful in validating the clusters when a number of cluster sets for the same dataset exist. Rinzivillo et al. proposes a visually guided clustering called progressive clustering [RPN*08], where the clustering is done with different distance functions in successive steps. In hierarchical clustering explorer [SS02], Seo and Shneiderman use an interactive dendogram, coupled with a color mosaic to represent clustering information in a linked visualization. They propose a cluster comparison view where two clustering results can be compared. However, their method is only suited for clusters of static data. In a recent study, Lex et al. introduce the MatchMaker [LSP*10], visualizing and comparing multiple groups of dimensions to represent cluster memberships. Their cluster visualization method is similar to our temporal cluster view, however, their solution does not provide information on the structural quality of clusters over time. Moreover, their method is designed for static clusters only. In the MultiClusterTree [VLL09], Long and Linsen discuss how clusterings are utilized to analyze multidimensional data. They use a radial layout, linked with several other views to explore hierarchical clusters. Telea and Auber [TA08] visualize changes in code structures using a flow layout where they try to identify steady code blocks and when certain splits in the code occur.

Cluster analysis of temporal data – One of the earliest works on cluster-based visualization of temporal data is by Wijk and Selow [VWVS99], where they cluster time-series data and visualize them on a calendar. Interactive clustering of trajectory data is discussed in a paper by Andrienko et al. [AAR*09], where they describe a user-driven clustering methodology. They use graphical summaries of trajectory clusters to indicate the number of cluster members. These summaries are sufficient when the analyst is interested in changes of the cluster sizes only. In an application of molecular dynamics analysis, Grottel et al. [GRVE07] use interactive visual tools to analyze clusters. The authors introduce the concept of flow groups and a schematic view, which displays cluster evolution over time. In a recent study, Rubel et al. [RWH*10] introduce a framework that integrates clustering and visualization for the analysis of 3D gene expression data. The authors integrate the data clustering for 3D gene expression analysis into their PointCloudXplore visualization tool. The approach in this study is application oriented, limiting a utilization in other fields. Self organizing maps (SOM) have been utilized in a recent study by Andrienko et al. [AAB*10]. They propose the interactive utilization of SOMs that are integrated in a visual analysis framework. Their solution aims to discover spatiotemporal relations by analyzing the temporal evolution of a spatial situation and the distribution of temporal changes sequentially.

Visual representations of temporal data – In this paper, we provide visual a representation of the structural changes of temporal clusters. There is a large number of studies on how to represent temporal data in visualization [WAM01, Moe05]. One of the important studies which represents temporal changes visually is the ThemeRiver [HHWN02] by Havre et al. The authors provide a visual representation of thematic changes in document collections over time. The ThemeRiver visualizes a single value per item and proposes a cumulative representation for each time step. In our temporal signatures, however, we encode a number of temporally varying statistics that are not suitable for a cumulative visualization due to their different scales.

In this paper, we extend the state of the art in the visual analysis of temporal clusters with the temporal cluster view, that integrates temporal clusters into interactive visual analysis procedures, and temporal signatures that visualize the temporal structure of clusters.

3. Overview

The proposed solution for analyzing temporal clusters is based on a new temporal cluster view (in the following just "cluster view") and temporal signatures. Firstly, we introduce the cluster view, that visualizes the quality of clusters together with structural changes that are related to item-cluster and cluster-cluster relationships. Secondly, we present temporal signatures, which are visual summaries of the statistical properties of clusters over time. The variations of these statistical properties reveals structural changes in groups of items.

These two views are utilized in an interactive visual analysis (IVA) cycle to analyze temporal clusters. Prior to the analysis, the analyst constructs a set of temporal clusters using a clustering algorithm. Information from the cluster view and the temporal signatures are combined with information on properties of items as provided by conventional views. The resulting insight is used to interpret and/or validate the clusters. This analysis is performed iteratively until sufficient clusters and insight in group relations is achieved. Fig. 2 is an overview illustration of our solution.

We present our solution in an IVA framework where we incorporate different types of linked views: histograms, scatterplots, parallel coordinates, (for regular variables), and functions graphs and animated scatterplots for temporal variables. In order to update these temporally varying views synchronously, we use a global time parameter τ. We define the dataset of independent variables (items) as $O = \{o_1, \ldots, o_n\}$, where each item has a set of $m = p + q$ dependent values $F(o_i) = [f_1(o_i), \ldots, f_p(o_i), g_{p+1}(o_i,t), \ldots, g_{p+q}(o_i,t)]$. Here, f represents regular variables and g represents time-series which are defined over time interval $[0,t']$. We define a temporal cluster c_i as:

$$c_i = \{I_{c_i}, T_{c_i} : I_{c_i} \subseteq O, T_{c_i} = [t_0,t_1], 0 \leq t_0 \leq t_1 \leq t'\} \quad (1)$$

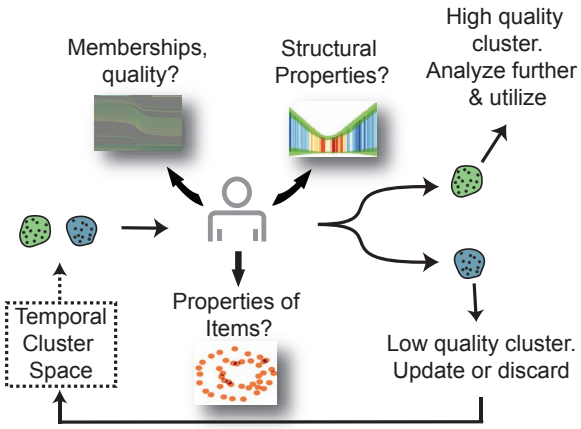

Figure 2: *An overview of our approach. A subset of temporal clusters are analyzed using our techniques and conventional IVA tools in terms of their structural changes and quality variations. Plausible clusters are analyzed to derive more insight on data. Low quality clusters are updated or discarded.*

In order to obtain such clusters, the analyst first defines a time interval T and then uses a clustering algorithm to cluster the data in T. This clustering operation is performed k times using different time intervals and/or item subsets which are determined by the user. We refer to the set of clusters obtained at each such step as a *clustering* C_j and the set of all the clusterings as $U = \{C_0, .., C_k\}$ where C_j is defined as:

$$C_j = \{c_1, .., c_{n_j} : \forall c_a, c_b (T_{c_a} = T_{c_b} \wedge c_a \neq c_b \Rightarrow c_a \cap c_b = \emptyset)\} \quad (2)$$

with n_j as the total number of clusters in C_j. Additionally, we do not necessarily expect C_j to include all the items in O, i.e., $\bigcup_{c \in C_j} \subseteq O$. Note that in a clustering, there are no overlapping clusters in terms of their items. However, it is possible that temporal spans of clusterings can overlap. In this paper, we use both hierarchical and k-means clustering [TSK06]. As these algorithms are originally developed for static data, we modified the distance measures as suggested by Liao [Lia05]. Our solution is well-suited to temporal versions of hierarchical and partitioning clustering algorithms, as they operate on distances between items. However, there exist also other algorithms which operate on densities and statistical models [Lia05]. To generalize our approach to a wider-variety of algorithm results, different quality metrics needs to be included into the analysis procedure.

In our framework, we utilize a brushing mechanism which is similar to *composite brushing* as proposed by Allen and Ward [MW95]. We extend this mechanism with selections over time. A brush $b = \{I, T\}$ is composed of an item selection, I ($I \subseteq O$), and a time interval selection, T ($[t_0,t_1]$). Each brush is combined with existing brushes by a Boolean operator S with $S \in \{\cup, \cap, \neg\}$, where $\cup$ represents the union,

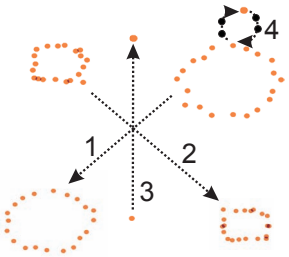

Figure 3: *In this artificial dataset, two groups move towards each other following the paths 1 and 2. One point follows path 3 and one item shortly gets away from its group (4).*

$\cap$ represents the intersection and $\neg$ represents the not operator. The result of this combination is a composite brush B, which is computed "in parallel" as the user makes brushes. Individual brushes b_i are combined into composite brushes B_i using the selected S by $B_i = S(B_{i-1}, b_i)$ starting with $B_1 = S(b_0, b_1)$. For simplicity, in the following, we denote the final set of brushed items as $B_L = \{I_L, T_L\}$. Note that, our definitions of a brush and a cluster (1) is the same, i.e., $b = \{I, T\} = c$. This enables the interpretation of clusters directly as brushes in our system. Due to the fact that non-continuous selections with respect to time would cause an additional complexity in the temporal analysis and related calculations, $\cup$ operator on time results in a single continuous time interval. The resulting time interval encapsulates both input intervals, i.e., $[t_0, t_1] \cup [t_2, t_3] = [min(t_0, t_2), max(t_1, t_3)]$. One other exception in the brushing mechanism is related to the $\cap$ operator in the temporal cluster view. In this view, when two brushes are combined using $\cap$, the item groups are intersected as expected with $\cap$. The temporal spans, however, are joined using $\cup$. This modification enables the use of the $\cap$ operator between clusters defined over non-overlapping temporal spans.

In order to demonstrate our approach in the following, we consider an artificial dataset (Fig. 3). In this dataset, two groups, composed of 20 points each, merge and split at certain points in time. There is a point that moves vertically from the bottom to the top. Additionally, one point shortly gets away from its group and returns back at the first half of the sequence. Prior to the analysis of this dataset, a number of clusterings are added to U. In order to avoid extra complexity in the analysis, we use all the items in consecutive clustering operations.

4. The Temporal Cluster View

The proposed temporal cluster view enables the visual exploration of clusters which are defined over different time intervals. It visually depicts how cluster memberships evolve over time. Moreover, it encodes cluster quality metrics and enables cluster level selections.

In the cluster view, each vertical axis visualizes a clustering C_k, where k indicates the order of the clustering in the view, i.e., for the leftmost axis, $k = 1$ (Fig. 4 a). Each *rectangle* on an axis corresponds to cluster c_i^k in C_k and each curve between the axes represents a single data item, o_i. When the user selects a cluster c in this view, I_c and T_c are handled by the selection mechanism as any other brush b with the above mentioned exception related to the $\cap$ operator.

We visualize the temporal span of clusters in order to link this view to the other temporally updating views. In Fig. 4 a, five clusterings C_{1-5}, performed on different time intervals, are visualized together with their temporal span on top. A black cursor is displayed at the top of the view to indicate τ. Temporal span of the clusterings, which are defined at τ, are highlighted by a saturated red color at the top of the view, e.g., C_2 in Fig. 4 a. Here, brushes b_1 and b_2 are combined using the $\cap$ operator, selecting the intersection of the items and the union of the temporal spans.

In order to encode information about the structural quality of clusters, we utilize the *silhouette coefficient* [Rou87], which is a popular method in data mining for evaluating the structural quality of clusters. Silhouette values s_i^k are computed per each item of cluster c_i^k and they are in the range $[-1, 1]$. Items close to cluster centers have higher values, and items on the borders of a cluster with close neighboring clusters have values close to 0. Moreover, when an item has a silhouette value close to -1, this item is wrongly placed in this cluster as an artifact of the clustering algorithm. In cluster view, we use silhouette values to color code curves and cluster rectangles. The color coding map, extracted from ColorBrewer [Bre09], is included in Fig. 4 b. The color of a single curve is interpolated between s_i^k and s_i^{k+1} and each cluster rectangle is colored according to the average of the s_i^k values of its members. Here, green colored curves and/or *rectangles* represent high-quality clusters (with respect to silhouette values).

In the cluster view, ordering is crucial for the ease of interpretation. Firstly, we order clusterings C_k according to the "start" of their time intervals T_{C_k}. Secondly, the c_i^k on each axis are ordered with a greedy algorithm in order to minimize overlapping curves between clusters. This ordering starts with the first clustering C_1 placed randomly. The algorithm then continues with the bottom-most cluster c_1^1 of C_1 and finds the cluster $x \in C_2$ which has the biggest overlap with c_1^1, i.e., $arg\,max_{x \in C_2} \left| c_1^1 \cap x \right|$. Then x is placed to the first available position on the second axis. The algorithm continues with c_2^1 and traverses all the clusters on the first axis. The same procedure is then applied for all the axes up to C_{n-1} where n is the number of axes. This crossing minimization problem is a well-known problem called "two layer crossing reduction problem" and more optimized solutions exist in literature [KW01]. Although it does not provide the optimum solution, we use the presented greedy algorithm due to its low computational complexity and its sufficient out-

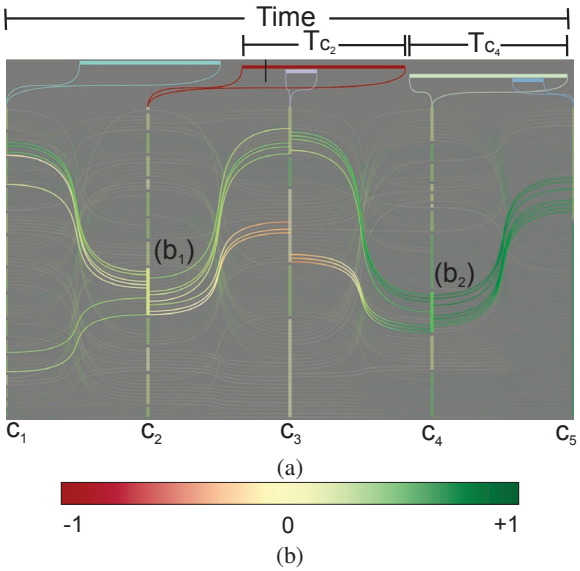

Figure 4: *(a) Five clusterings visualized in the cluster view. The temporal span of each clustering is visualized on top. Brushes b_1 and b_2 are made to select two clusters. (b) Color coding for silhouette values.*

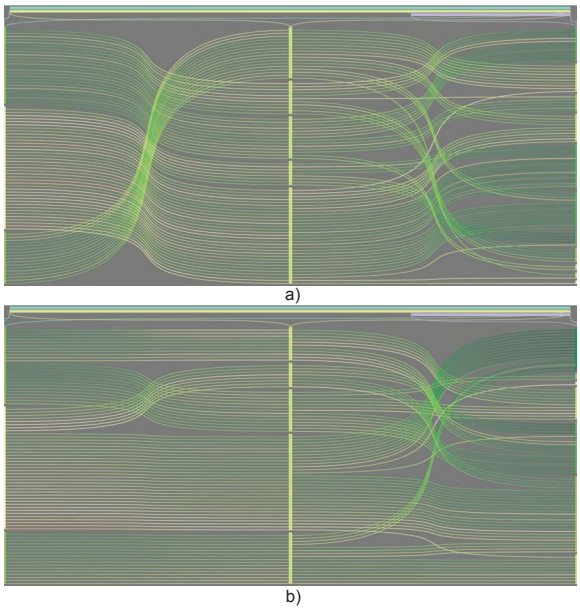

Figure 5: *Ordering cluster view improves the overall perception of cluster quality. Before (a) and after ordering (b).*

come for the requirements of our solution. Finally, we order the items in the clusters. All the members of the clusters are first grouped according to the *branches* between C_k and C_{k+1}, where a branch represents overlapping items between two clusters, i.e., $c_i^k \cap c_j^{k+1}$. As the final step in this ordering, all the items in a single branch are organized in an ascending order with respect to s_i^k values. The effect of ordering on the perception of cluster relations and cluster quality is illustrated in Fig 5.

Although our clustering definition (2) allows for items that are not members of any clusters, the clustering algorithms we use in this paper assigns all the items to clusters. In case of items which are not in a cluster (can be referred to as outliers), these items are grouped together and visualized just like any other cluster in the cluster view. If the analyst plans to focus on these outliers, this group of outlier items can be visualized in a distinctive color in the cluster view.

5. Temporal Signatures

In order to explore the structural changes in temporal clusters, we rely on a qualitative approach based on structural statistics, which is easy to interpret, calculate, and visualize. Fig. 6 demonstrates the proposed measures. We utilize a group coherence measure that is based on mutual distances between items in I_L for every time step in T_L. Note that I_L can consist of any group of items that are selected by the brush combinations in the framework. Here, we compute average distance boundaries, which can be thought of as computing the extent covered by points I_L—referred to as *cluster diameter* [ELL09]. The minimum average distance Min_{avg}^t and maximum average distance Max_{avg}^t are calculated for all time steps separately as follows:

$$Min_{avg}^t = \frac{\sum_{i=1}^{|I_L|} d_{min}^t(o_i)}{|I_L|}, \tag{3}$$

where $d_{min}^t(o_i) = min(\{d^t(o_i,o_j)|o_{i,j} \in I_L \wedge o_j \neq o_i\})$ and t represents a single time step. Max_{avg}^t is computed likewise with *max* instead of *min* in equation (3). Per each time step, we additionally compute the sum of number of items "closer" to each other than a distance threshold D. This number, which we refer to as *vicinity measure* V, describes the *compactness* (cohesion) of the group [TSK06]. D is a free parameter, which users can interactively change according to the Min_{avg} and Max_{avg} values. $V^t(D)$ is defined by:

$$V^t(D) = \sum_{i=1}^{|I_L|} \left|\{j|o_j \in I_L \wedge o_j \neq o_i \wedge d^t(o_i,o_j) < D\}\right|. \tag{4}$$

For equations (3) and (4), the Euclidean distance is preferred for $d^t(\cdot,\cdot)$, which is defined as: $d^t(o_i,o_j) = \sqrt{\sum_{k=1}^{q}\left(g_k(o_i,t) - g_k(o_j,t)\right)^2}$ where g are the temporal variables in our dataset. The selection of distance functions is an essential element of cluster analysis and the utilization of several distance functions can be found in the literature [STH*09]. Therefore, the distance function should be chosen to fulfill domain specific constraints.

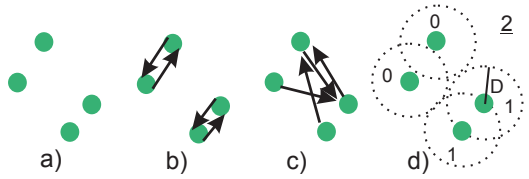

Figure 6: *For four 2D points (a), we compute minimum distances (b), maximum distances (c), and vicinity measure V (d). V is the sum of neighboring items within a sphere of radius D* $(0+0+1+1=2)$.

The temporal signature view computes the above defined metrics for the currently selected group of items (not necessarily from a cluster) over the selected time interval to construct the visualization. Fig. 7 (left) shows an example of such a temporal signatures view, where I_L contains all the items for the whole time span of the dataset. The upper bound represent maximum average distances, while the lower one represent minimum average distances. We also compute the standard deviations of these distances and render them in a transparent green band around the actual minimum and maximal values. Moreover, we utilize the space between the boundaries to display V values by color intensities. The saturated blue colors represent sparsely distributed items, while the saturated red colors represent packed items, i.e., higher number of neighboring items. The color scaling is done according to the minimum and the maximum values of V for the current I_L and T_L. In Fig. 7 (left), we can observe an instability between Min_{avg} and Max_{avg} values, where the band gets thinner in the middle as time progresses. This is due to the fact that the groups at t_0 cross each other at t_1 making the overall cluster diameter smaller.

Both standard deviations, $stdev(Min_{avg})$ and $stdev(Max_{avg})$, encodes cluster homogeneity. In Fig. 7, we select first $I_L = c_1$, then $I_L = c_2$, and eventually $I_L = c_1 \cup c_2$ over $T_L = [t_0, t_1]$. The signature of cluster c_1 indicates a high quality cluster due to the stable values of the metrics. However, cluster c_2 contains an outlier (Fig. 3-4), that is recognized through the peaking standard deviations. In general, $stdev(Min_{avg})$ reveals outliers. $stdev(Max_{avg})$ is mainly associated with cluster homogeneity where lower values identify tightly packed items or groups of such tightly packed items. For instance, although group $c_1 \cup c_2$, separates at t_0, the resulting $stdev(Max_{avg})$ values do not vary when c_1 and c_2 merges at t_1, except for the outlier in c_2.

For all the views in Fig. 7, we specify $D = max\{Min_{avg}\}$, which means that there is a number of items above D for all the time steps. This choice of D reveals only the most compact configuration of the items over the whole time interval. In the rightmost signature view in Fig. 7, it can be seen that items are in the most compact form at t_2 (saturated red color) where c_1 and c_2 merges.

Instead of arbitrary groups of items, the analyst can prefer to directly brush clusters. In this case, the signature view enables the user to perform a number of analysis tasks on clusters:

- A single cluster can be visualized to evaluate its temporal structural variations.
- A number of clusters can be brushed using S operators to explore the resulting group's behaviors.
- While a single cluster is selected, the temporal selection (T_L) can be expanded using other brushes. The resulting signature view visualizes how this cluster behaves over time intervals where it is not defined.

6. Temporal Cluster Analysis Procedures

Temporal-cluster analysis aims to find a plausible set of clusters and understand the structural variations of these clusters. The analysis starts with visualizing the selected clusterings in the cluster view and continues with selecting a number of clusters and investigating the corresponding temporal signatures. As a result of the interpretations of these two views, the analyst draws one of these conclusions; validate a cluster, update the temporal span of a cluster or discard a cluster. In order to draw such conclusions, interpretation of silhouette values and discovering where structural changes (like splitting and merging) take place is quite important.

Interpreting silhouette values – Silhouette values are higher when the clusters are well-separated and more coherent. Therefore, in regions with not so apparent clusters (i.e., where the distribution of items is more uniform), the silhouette values are generally close to zero or even below zero. In Fig. 8, we can see a clear example of such a situation. Here, the example dataset (Fig. 3) is clustered over consecutive time intervals (C_{1-6}). As the distribution of items where two groups meet is quite uniform, we see that the colors of items and clusters turn to yellow. However, near the beginning and at the end of the sequences, the overall cluster quality is high, and this is clearly visible from the colors of C_1 and C_6. This observation yields to the fact that clusters performed over the merging interval are lower in structural quality and therefore, have to considered with more care when further analysis is performed on them.

Merging and splitting – Two of the important behavior of clusters are merging and splitting. To analyze these behaviors, we firstly brush a cluster by $\cap$ operation, which may represent a cluster that is about to split or to be created as a union of several other clusters. Secondly, we observe the accompanied temporal signatures view, which reveals this structural tendencies.

In Fig. 9 a, we visualize three sequential clusterings, C_1, C_2 and C_3. We brush a cluster (b_1) in C_2 by $\cap$ brush and by brushes b_2 and b_3 ($\cup$) we extend time selection T_L to contain also time intervals of C_1 and C_3. This extension of time interval is crucial to show the behavior of cluster b_1 in C_1

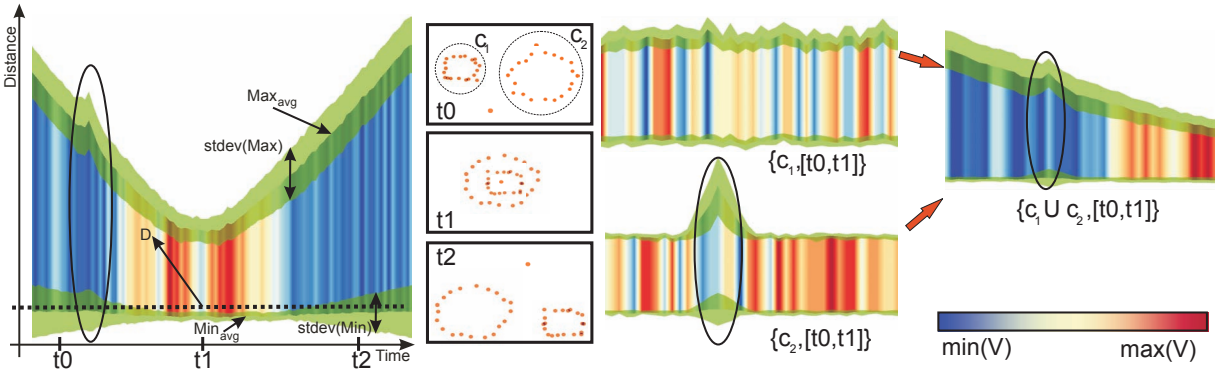

Figure 7: *Left: Temporal signatures view. The upper bound represents maximum average distance, the lower represents minimum average distance and the vicinity measure is represented with the color map depicted on the bottom right. The dotted line represents the threshold distance D. The standard deviation is rendered through the transparent green color. Circles mark changes due to movement 4 in Fig. 3. Right: Signature views computed for clusters c_1, c_2 and $c_1 \cup c_2$ over time interval $[t_0, t_1]$.*

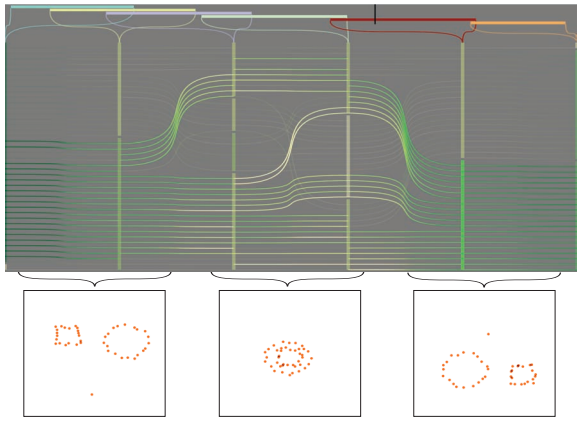

Figure 8: *Variation of silhouette values. Group structures change as items move over time. These variations are clearly visible in cluster view by observing the color changes.*

and C_2. The signatures view is then automatically updated for $I_L = I_{C_2}$ and $[T_{C_1} \cup T_{C_3}]$. A notable tendency is that the band between the Min_{avg} and Max_{avg} gets smaller towards T_{C_2} (*cluster merging*) and gets large again at T_{C_3} (*cluster splitting*). We can additionally observe that I_L has the most compact form where both groups merge and a sparse form where the groups are separated by observing V values.

To generalize, we follow a set of informal rules in evaluating the clusters using our views:

- Items in a cluster should not have many branchings in cluster view
- Cluster rectangle and item curves should be in saturated green
- In a signature of a cluster, values of Min_{avg} and Max_{avg} and the thickness of the band between them should not deviate
- Signatures should mostly contain red values in the band (i.e., high V values)

7. Case Study: Analysis of Molecular Dynamics of Mixed Lipid Bilayers

Molecular modeling of biological membranes is one of the application fields where analysis of temporal clusters is particularly useful. Cell membranes separate the interior of cells from the environment and are mostly constituted of a mixture of different lipids. The lipids can form microdomains or clusters with other membrane components. Such microdomains are relevant for signal transduction or cell apoptosis to name but a few [FSH10]. Lipid bilayers are widely used to model and study cell membranes, and molecular dynamics (MD) simulations are utilized as powerful tools to describe their atomic structure and dynamic behavior. These simulations run on a mixture of different types of lipids that form different cluster sets. These lipid clusters can lead to inhomogeneity in biological membranes [BR10].

Here we use a dataset obtained from MD simulation of a mixed lipid bilayer [BR10], constituted of DMPC (dimirystoilphosphatidylcholine) and DMPG (dimirystoilphosphatidylglycerol) lipids composed of 1640 time steps. Each lipid is represented by one particle, localized at the position of the phosphorus atom. Additionally, we work on a set of clusterings $\{C_1, \ldots, C_n\}$ that are computed as the final step of the simulation phase.

Our aim here is to evaluate clusters by their stability over time. In case of a plausible cluster (with respect to cluster view and to signatures view), we perform additional IVA analyses to specify the time span where the cluster preserves its structure.

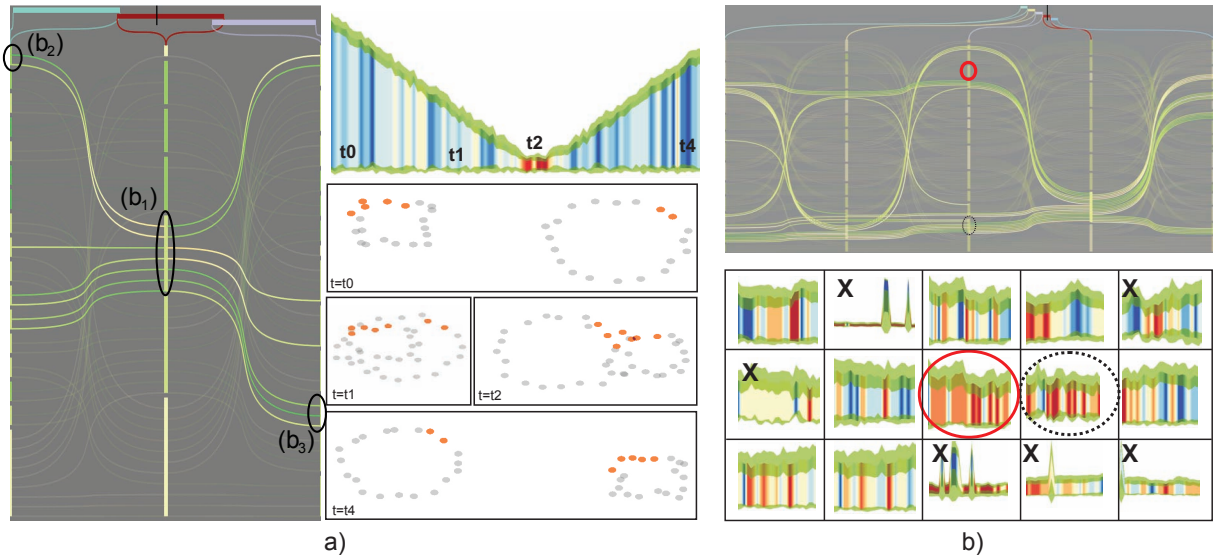

Figure 9: *a) Cluster merging-splitting behavior. A cluster is selected with b_1 and the time selection is enlarged by brushes b_2 and b_3. Merging occurs around the smaller band in the middle, which gets larger at end of the sequence due to splitting in signature view. b) Searching for a plausible cluster. Two good signatures are identified (circles). The dashed circle is discarded due to its structural instability in cluster view (shown with the selection on the right). The red circled cluster is picked for further analysis. Moreover, the observed signatures allow to discard clusters (**X**) according their structure.*

We start the analysis by displaying the clusterings in the cluster view. Then we assess the cluster quality, firstly, by brushing individual clusters by $\cap$ operation according to silhouette values, and secondly, assessing the cluster coherence via the signature view. Here, we use the set of rules described in Section 6. Fig. 9 b displays a set of signatures for the observed clusters C_{1-5} defined over sequential time intervals $T_{C_{1-5}}$.

In Fig. 9 b, although the signature for the cluster marked with dotted circle represents a good cluster; the cluster structure over time is not stable due to branching in cluster view. Therefore, this cluster is not picked for further analysis. Discarded clusters are marked with an X in the figure. Nevertheless, we found a cluster (marked with a red circle in Fig. 9 b) that has a plausible signature and exhibits a stable structure in the neighboring clusterings. We continue our analysis with this cluster c in C_3 (Fig. 10). As the next step, we enlarge the time selection, from $T_L = T_{C_3}$ to $T_L = [T_{C_1} \cup T_{C_5}]$. The corresponding signature Fig. 10 (left-bottom) depicts the stability of cluster c even for the remaining intervals. The stability is observed by the band width between minimum average distance Min_{avg} and maximum average distance Max_{avg}. The group extend is preserved over T_L since $stdev(Max_{avg})$ has the same width for the whole time. However, $stdev(Min_{avg})$ exhibits certain instabilities which are caused by oscillation movements of cluster boundary lipids that gets away from the group for a few time steps. Additionally, we continue by extending time interval T_L with a brush on time domain (not shown in the figure) to analyze how stable this group is over a larger time interval. With this update, we observe that the signature changes rapidly for latter regions (Fig. 10 (top-right)). This limits the time extend of this cluster to the first peak (arrow). However, later on, we can see that the vicinity values, depicted by colors, are close to red again, identifying that the same group is forming. Since we observe this region where the cluster can be defined, we add clustering C_6 for this region of interest. We see in Fig. 10 (bottom-right) that cluster c is formed again, even for this small interval.

Our collaborators working in the field of biomolecular modelling state that, in their previous work on a similar dataset [BR10] they faced many limitations in performing analysis on group behaviors. Due to the complexity of analyzing the clusters over time, they were doing the clustering on individual time steps and average the clustering properties over time. As they were not able to relate the structure of these separate clusterings, they were computing properties of them and analyze the changes of these values over time. These statistics involve basic properties like the number of clusters and the number of items in clusters at each time step. In their analyses, it was not possible to explore the behavior and quality of clusters over time. They state that our framework provides significant improvements in the analysis of MD simulations of lipid bilayers. The proposed framework enables the discovery of grouping behaviors which can lead to new hypotheses on the relations of lipids in lipid bilay-

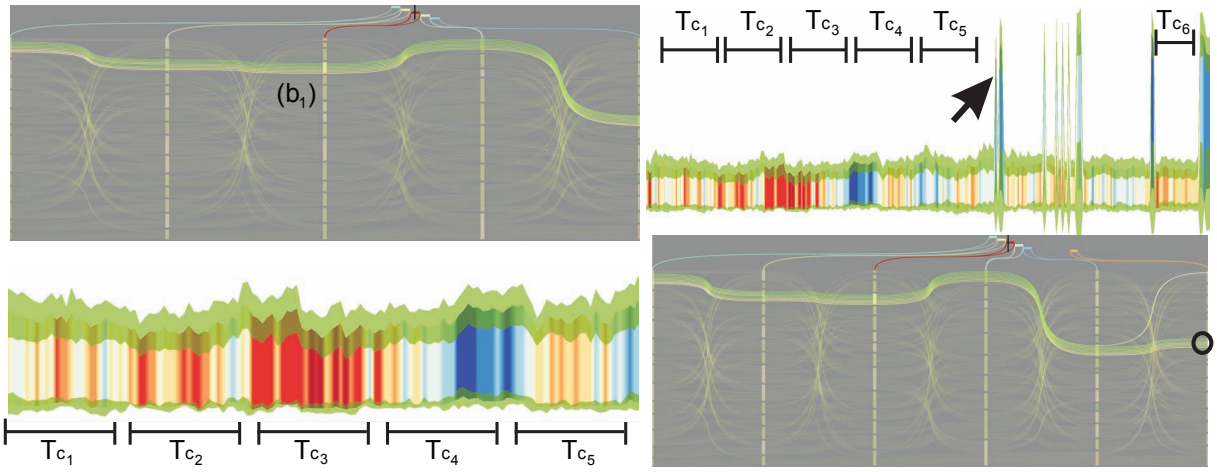

Figure 10: *Lipid cluster development. Top left: Coherent cluster $c = b_1$ in all clusterings, C_{1-5}. Bottom left: The signature view for c with extended time interval to showcase signatures in remaining clustering intervals $T_L = [T_{C_1} \cup T_{C_5}]$, where it expresses high stability. Top right: We extend T_L to search for "existing" boundaries (arrow) for cluster c, where we mark another coherent interval T_{C_6}. We add cluster C_6, where we observe that items in b_1 reforms cluster c again at T_{C_6} (circle).*

ers. During this case study, we came across a number of additional analysis tasks like: identifying the threshold deviations for the "good" lipid clusters, analysis of vanishing clusters and determining the time when the overall system stabilization takes place. These are potential tasks where our analysis framework can be utilized. In general, our collaborators find the procedure to be faster, more powerful and more reliable than traditional approaches which are usually based on distance criteria applied to each frame of the sequence.

8. Conclusion

In this paper, we introduce two novel visualization techniques for the interactive visual analysis of temporal clusters. We firstly introduce cluster view, which interactively visualizes a number of clusters defined on temporal intervals. This view visualizes the variation of the structural quality of clusters by representing the changes of silhouette coefficients. Cluster view visualizes the temporal span of clusters in order to enable the exploration of clusters over time. Secondly, we present temporal signatures which are visual representations of the structure of a group of items over time. This view encodes a number of time-varying statistical properties of a group to depict its structural transformations. We show how these views enable an intuitive analysis of temporal clusters, where the analyst is able to determine the validity of the clusters and interpret the relations that cause structural changes in clusters. To the best of our knowledge, our solution is the first interactive visual approach to analyze the structural changes in cluster-cluster and item-cluster relations of temporal datasets.

We integrated our visualizations into an IVA environment where we performed visual analysis of temporal clusters. Cluster view enables cluster level interactions and when used in combination with temporal signatures view, it provides a mechanism to explore temporal clusters in terms of their structural properties. We describe analysis procedures which enables the analyst to explore the quality of clusters over time and explore the structural changes exhibited by clusters. As a consequence of these analyses, the clusters are either validated, updated or discarded. The analyst then continues with the further analyses of high quality clusters.

We evaluated our methodologies on the analysis of molecular dynamics simulation, where the analyst is trying to build hypotheses on the grouping behaviors of lipid-bilayers. We show that our methods reveals certain groups which exhibit stable behavior over distinct time intervals. Such behavior patterns provides the basis to make hypotheses on the behavioral properties of lipid bilayers.

As a future work, we plan to extend our temporal signatures with more robust statistics and different quality metrics, which can provide deeper insight on the structure of groups of items. Another future direction is to create abstract representations of the structural changes and encode them in the form of an event based visualization system.

Acknowledgments

NR acknowledges funding from the Bergen Research Foundation (BFS; Bergens Forskningstiftelse) and the University of Bergen. We thank Torben Broemstrup for molecular dynamics simulation data.

References

[AAB*10] ANDRIENKO G., ANDRIENKO N., BREMM S., SCHRECK T., VON LANDESBERGER T., BAK P., KEIM D.: Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns. *Computer Graphics Forum 29*, 3 (2010), 913–922. 2

[AAR*09] ANDRIENKO G., ANDRIENKO N., RINZIVILLO S., NANNI M., PEDRESCHI D., GIANNOTTI F.: Interactive visual clustering of large collections of trajectories. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on* (2009), IEEE, pp. 3–10. 2

[BR10] BROEMSTRUP T., REUTER N.: Molecular Dynamics Simulations of Mixed Acidic/Zwitterionic Phospholipid Bilayers. *Biophysical journal 99*, 3 (Aug. 2010), 825–833. 7, 8

[Bre09] BREWER C. A.: http://www.colorbrewer.org/, 2009. 4

[ELL09] EVERITT B. S., LANDAU S., LEESE M.: *Cluster Analysis*, 4th ed. Wiley Publishing, 2009. 5

[FSH10] FAN J., SAMMALKORPI M., HAATAJA M.: Formation and regulation of lipid microdomains in cell membranes: Theory, modeling, and speculation. *FEBS letters 584*, 9 (2010), 1678–1684. 7

[GRVE07] GROTTEL S., REINA G., VRABEC J., ERTL T.: Visual verification and analysis of cluster detection for molecular dynamics. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1624–1631. 2

[HHWN02] HAVRE S., HETZLER E., WHITNEY P., NOWELL L.: Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics 8* (January 2002), 9–20. 3

[KW01] KAUFMANN M., WAGNER D.: *Drawing graphs: methods and models*. Springer Verlag, 2001. 4

[Lia05] LIAO W.: Clustering of time series data–a survey. *Pattern Recognition 38*, 11 (2005), 1857–1874. 1, 3

[LSP*10] LEX A., STREIT M., PARTL C., KASHOFER K., SCHMALSTIEG D.: Comparative analysis of multidimensional, quantitative data. *Visualization and Computer Graphics, IEEE Transactions on 16*, 6 (2010), 1027 –1035. 2

[Moe05] MOERE A.: Time-varying data visualization using information flocking boids. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on* (2005), IEEE, pp. 97–104. 3

[MW95] MARTIN A. R., WARD M. O.: High dimensional brushing for interactive exploration of multivariate data. In *VIS '95: Proceedings of the 6th conference on Visualization '95* (Washington, DC, USA, 1995), IEEE Computer Society, p. 271. 3

[Rou87] ROUSSEEUW P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics 20* (1987), 53–65. 2, 4

[RPN*08] RINZIVILLO S., PEDRESCHI D., NANNI M., GIANNOTTI F., ANDRIENKO N., ANDRIENKO G.: Visually driven analysis of movement data by progressive clustering. *Information Visualization 7*, 3 (2008), 225–239. 2

[RWH*10] RUBEL O., WEBER G., HUANG M.-Y., BETHEL E., BIGGIN M., FOWLKES C., LUENGO HENDRIKS C., KERANEN S., EISEN M., KNOWLES D., MALIK J., HAGEN H., HAMANN B.: Integrating data clustering and visualization for the analysis of 3d gene expression data. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on 7*, 1 (2010), 64 –79. 2

[SGM08] SHARKO J., GRINSTEIN G., MARX K.: Vectorized radviz and its application to multiple cluster datasets. *IEEE transactions on Visualization and Computer Graphics* (2008), 1444–1427. 2

[SS02] SEO J., SHNEIDERMAN B.: Interactively exploring hierarchical clustering results. *IEEE Computer 35*, 7 (2002), 80–86. 2

[SS05] SEBORG A., SINGHAL A.: Clustering multivariate time-series data. *Journal of chemometrics 19*, 8 (2005), 427. 1

[STH*09] SHI K., THEISEL H., HAUSER H., WEINKAUF T., MATKOVIC K., HEGE H., SEIDEL H.: Path line attributes-an information visualization approach to analyzing the dynamic behavior of 3d time-dependent flow fields. *Topology-Based Methods in Visualization II* (2009), 75–88. 5

[TA08] TELEA A., AUBER D.: Code flows: Visualizing structural evolution of source code. *Computer Graphics Forum 27*, 3 (2008), 831–838. 2

[TSK06] TAN P., STEINBACH M., KUMAR V.: *Introduction to data mining*. Pearson Addison Wesley Boston, 2006. 2, 3, 5

[VLL09] VAN LONG T., LINSEN L.: MultiClusterTree: Interactive Visual Exploration of Hierarchical Clusters in Multidimensional Multivariate Data. In *Computer Graphics Forum* (2009), vol. 28, John Wiley & Sons, pp. 823–830. 2

[VWVS99] VAN WIJK J., VAN SELOW E.: Cluster and calendar based visualization of time series data. In *infovis* (1999), Published by the IEEE Computer Society, p. 4. 2

[WAM01] WEBER M., ALEXA M., MÜLLER W.: Visualizing time-series on spirals. In *proceedings of the IEEE Symposium on Information Visualization* (2001), Citeseer, p. 7. 3

[WSH06] WANG X., SMITH K., HYNDMAN R.: Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery 13*, 3 (2006), 335–364. 1