# Integrating Cluster Formation and Cluster Evaluation in Interactive Visual Analysis

Cagatay Turkay*
Department of Informatics
University of Bergen

Julius Parulek†
Department of Informatics
University of Bergen

Nathalie Reuter‡
BCCS
University of Bergen

Helwig Hauser§
Department of Informatics
University of Bergen

## Abstract

Cluster analysis is a popular method for data investigation where data items are structured into groups called clusters. This analysis involves two sequential steps, namely cluster formation and cluster evaluation. In this paper, we propose the tight integration of cluster formation and cluster evaluation in interactive visual analysis in order to overcome the challenges that relate to the black-box nature of clustering algorithms. We present our conceptual framework in the form of an interactive visual environment. In this realization of our framework, we build upon general concepts such as cluster comparison, clustering tendency, cluster stability and cluster coherence. Additionally, we showcase our framework on the cluster analysis of mixed lipid bilayers.

**CR Categories:** I.3.m [Computing Methodologies]: Computer Graphics—Miscellaneous; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Clustering

**Keywords:** Visual Analysis Models, Visual Knowledge Discovery, Data Clustering, Bioinformatics Visualization.

## 1 Introduction

Cluster analysis divides data into groups (clusters) where data items within a group are similar with respect to certain criteria. Usually, data items are clustered using solely the information which is available in the data which represents the items and their relations. Clusters provide the analyst with a grouping structure without providing any information on why they exist and which properties they have (e.g., whether the clustering is stable) [Tan et al. 2006]. Conventionally, cluster analysis involves two consecutive steps; *cluster formation* and *cluster evaluation*. Cluster formation is a black-box operation where the user specifies a clustering algorithm together with a set of parameters and gets an according clustering. Here, we refer to clustering as the entire set of clusters. Usually, the formation step is followed by an evaluation phase where the user decides whether she is satisfied with the clustering, or not. If the results are implausible, the process is carried out again with a different parameter set and/or algorithm.

*e-mail: Cagatay.Turkay@uib.no
†e-mail:Julius.Parulek@uib.no
‡e-mail:nathalie.reuter@mbi.uib.no
§e-mail:Helwig.Hauser@uib.no

Assessing a clustering's quality and fine tuning the clustering algorithms are complex tasks due to the following facts [Tan et al. 2006]. Firstly, the relations in the data that eventually lead to a clustering vary from domain to domain. This makes it hard to generalize and formalize the definition of what a valid cluster is. Secondly, clusters do not usually reveal any implicit information on data relations, making them harder to be interpreted. Thirdly, clustering algorithms are highly dependent on their parameters and often these parameter sets do not offer a good basis for the analyst to steer the analysis using her domain knowledge. There is a certain need for mechanisms to enhance cluster analysis. These mechanisms should include a stronger utilization of the analyst's domain knowledge in cluster analysis together with methods for the interactive analysis of raw data (together with the clusters). Such mechanisms would not only lead to more satisfactory clusterings but also provide more insight into the underlying relations in data. This insight eventually increases the confidence of the expert on cluster analysis results.

As an illustration of a situation where integration of the expert in cluster analysis is required, let us assume a demonstrational case (Fig. 1), where points are scattered on a 2D plane without any structurally apparent clustering. With a slight change in input parameters, two runs of a clustering algorithm can result in different clusterings (Fig. 1b,c). After the evaluation phase, the analyst can reckon that both clusterings are unsatisfactory. However a combination of both results, which is not necessarily the outcome of any clustering algorithm, could be a satisfactory clustering (Fig. 1d). In order to resolve this ambiguity, the analyst needs to steer the cluster formation process by using a combination of her domain knowledge and the insight gained throughout the analysis.

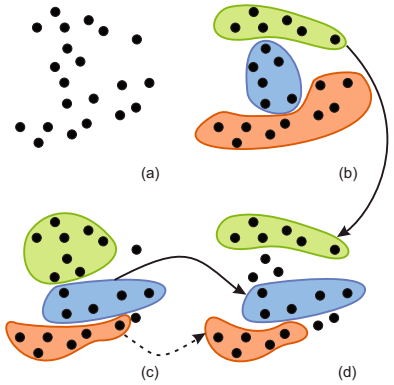



Figure 1: Ambiguity in the cluster analysis of a set of 2D points a) Initial set of points; b),c) Two possible clustering results; d) The resulting clusters obtained by a combination of the original two clusterings, where regarding one cluster there is a need to change the cluster by removing some points (dashed arrow).

In data mining and cluster analysis, research has been done on computational techniques to achieve successful cluster analysis. Jain [Jain 2010] provides a detailed and up-to-date description of the problems and techniques in cluster analysis. A basic and fun-

damental subset of these techniques relate to *clustering tendency*, *cluster comparison*, *cluster stability* and *cluster cohesion*. Clustering tendency reveals if there is a non-random structure in the selected data sample [Smith and Jain 1984]. An apparently structured sample would lead to a successful clustering much more than a sample with almost a random structure. Ideally, cluster analysis begins with the evaluation of the clustering tendency of the investigated data. Cluster comparison is a clear requirement of cluster analysis owing to the fact that clustering algorithms are often quite stochastic and parameter dependent [Jain 2010]. Cluster analysis does not usually rely on a single result of a particular algorithm. Analysts make a number of different clusterings with different parameters (and algorithms). Intuitive and interactive mechanisms to compare clusters need to be a part of any cluster analysis process. Cluster stability and cluster cohesion are important criteria to evaluate the validity of a cluster. A cluster can be considered stable when its members are generally clustered together in different clusterings [Lange et al. 2004]. Cluster cohesion, on the other hand, depicts how tight the items are in a single cluster [Tan et al. 2006]. It is possible to observe additional structures in a cluster with low cohesion. Analysts can reckon that an unstable and/or an incoherent cluster is not valid and that it requires a refinement. Conventional cluster analysis employs these techniques separately in the steps of cluster formation and cluster evaluation in order to achieve high quality clusterings [Tan et al. 2006]. However, we are not aware of any solution which integrates these techniques in an interactive and iterative analysis procedure.

To achieve this, we describe a framework that tightly integrates cluster formation and cluster evaluation in interactive visual analysis (IVA). In cluster formation we employ the explorative power of the human perception to discover subsets of the data with higher clustering *tendency*. In cluster evaluation, we utilize the expert's domain knowledge to *compare* and evaluate clusters in terms of their *stability* and *cohesion*.

In this paper, we also realize and present the proposed framework in the form of an interactive visual environment. In this environment, we incorporate conventional views as well as two specific views for the two aforementioned purposes. The conventional views are scatter plots, histograms and function graphs which are used in linking&brushing operations. The two more special views are a *cluster tendency view* to evaluate the suitability of data subsets for clustering and a *parallel cluster view* to interact with a number of different clusterings and to compare them. Importantly, in all the stages of the analysis, clusters are treated as an additional dimension of the actual dataset. This approach enables us to tightly integrate clusters in the IVA cycle.

Keim et al. [Keim et al. 2008] stated that an important task of visual analytics is to integrate the knowledge, explorative power and creativity of the human with the computational power of algorithms. We follow this research goal with the visual exploration and analysis of clusterings which not only enable experts to discover more reliable groupings in data but also provide information on why these groupings exist at all.

## 2 Related Work

Interactive techniques have proven to help analysts to manually refine and build clustering results. A hierarchical clustering and visualization algorithm, H-BLOB, is introduced by Sprenger et al. [Sprenger et al. 2000]. The authors propose a visual clustering approach which involves a two-stage procedure, where a hierarchical clustering is followed by a visualization, using blob objects, to reveal cluster shapes. Rinzivillo et al. use a visually driven technique called progressive clustering [Rinzivillo et al. 2008] where the clustering is done in successive steps using different distance functions. The authors show that the progressive clustering technique provides a convenient mechanism, where a user can selectively direct the algorithms to potentially interesting portions of data. Schreck et al. [Schreck et al. 2008] propose a framework to interactively monitor and control Kohonen maps to cluster trajectory data. In their paper, they state the importance of integrating the expert in clustering process to achieve suitable results.

Visualization has generally served as the final step of cluster analysis where it plays a critical role in enhancing the interpretation of clusters by enabling comparison and evaluation. Grottel et al. [Grottel et al. 2007] use interactive visual tools to analyze clusters in molecular dynamics. The authors introduce the concept of flow groups, which display cluster evolution over time, to validate the quality of clustering results. In a recent study, Rubel et al. [Rubel et al. 2010] introduce a framework that integrates clustering and visualization for the analysis of 3D gene expression data. Authors integrated the data clustering for 3D gene expression analysis into their PointCloudXplore visualization tool. The approach in this study is application oriented, therefore enables only a limited utilization. On the contrary, our framework can be applied to arbitrary multivariate and/or time varying datasets.

In *Hierarchical Clustering Explorer* [Seo and Shneiderman 2002], Seo and Shneiderman use an interactive dendogram coupled with a color mosaic to represent clustering information together with conventional visualizations. They also include a cluster comparison view where the user can compare two clustering results. In a recent study, Lex et al. introduce MatchMaker [Lex et al. 2010], where they visualize and compare multiple groups of dimensions. In their work, they provide a use-case where they use their methods to compare clusters. We enrich this cluster comparison capability by mechanisms to compare clusters not only on member items but also on quality. Vectorized radial visualizations are used in exploring different clustering results by projecting data records on a vectorized cluster space [Sharko et al. 2008]. This approach proves to be useful in validating the clusters when many different clusterings for the same dataset exist. An interactive dissimilarity matrix, presented by Bezdek and Hathaway [Bezdek and Hathaway 2002], was extended to analyze clustering results at different similarity level by Siirtola [Siirtola 2004]. Sharko et al. [Sharko et al. 2007] use heat maps called cluster stability matrices to visually analyze and reveal most 'stable' clusters in clustering results. In our approach, we enhance the cluster comparison capability in the above studies [Seo and Shneiderman 2002] [Sharko et al. 2008] by using IVA operations in the parallel cluster view. Moreover, we enhance and integrate the dissimilarity matrix [Bezdek and Hathaway 2002] visualizations with cluster comparison plot in an interactive visual analysis cycle to fill the gap between cluster evaluation and cluster formation in cluster analysis.

## 3 Integrating Cluster Formation and Cluster Evaluation

As mentioned in Section 1, our approach integrates cluster formation and cluster evaluation steps in an interactive visual analysis environment. Our conceptual framework is realized in an interactive visual analysis environment (Fig. 2) where we employ a view to explore cluster tendency (cluster tendency view) and a view to compare clusters and utilize them in IVA (parallel cluster view). In our approach, cluster analysis starts with the cluster formation phase. The first step in this phase is the exploration of a suitable domain

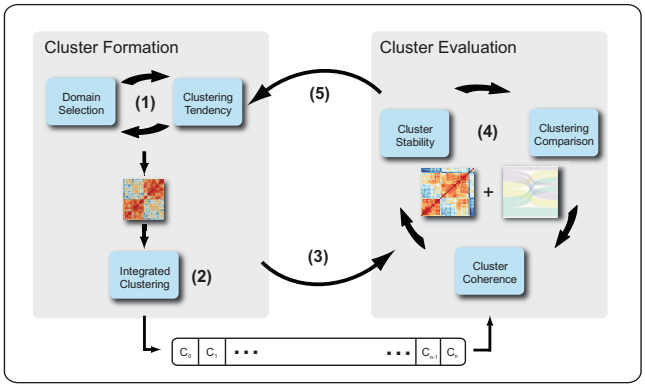



Figure 2: An illustration of our framework. Iterative exploration is performed to find suitable data subsets for clustering (1). Suitable selections are clustered and stored (2). Analysis continues with the evaluation phase (3). Evaluation is carried out in an IVA cycle by comparing clusters with respect to their stability and coherence (4). Analysis continues by incorporating the insight from the evaluation phase (5).

for a clustering (Fig. 2-1). Here, the domain refers to brushed items and a set of dimensions to use in clustering. The selected domains are evaluated with respect to their tendency for clustering by using the cluster tendency view. The selected domain is then clustered using one of the clustering methods and the resulting clusterings are stored (Fig. 2-2). Cluster analysis continues with the evaluation phase (Fig. 2-3). This phase is performed using the cluster tendency view and the parallel cluster view together with conventional visualization support (Fig. 2-4). In this phase, the parallel cluster view is used to assess cluster stability and to compare clusters. Additionally, when used in conjunction with the cluster tendency view, it provides a mechanism to discover cluster coherence. Cluster analysis is iteratively continued using the insight gained in the evaluation phase (Fig. 2-5).

In the realization of our conceptual framework we follow the data model presented by Konyha et al. [Konyha et al. 2006], which provides a suitable and flexible language for the analysis of arbitrary data types. We define the data set of independent variables (items) as $O = \{o_1, \ldots, o_n\}$, where each item has a set of $m = p+q$ dependent values $F(o_i) = [f_1(o_i), \ldots, f_p(o_i), g_{p+1}(o_i,t), \ldots, g_{p+q}(o_i,t)]$, where $f$ represents regular and $g$ represents temporal variables. The regular variables have a singular value for each $o_i$ and temporal variables change their values over a time interval. We define the cluster $c$ as $c \subseteq O$ and the clustering $C$ as a set of clusters $C \subseteq 2^O$, where the following clustering criteria hold:

$$\bigcup_{c \in C} \subseteq O \text{ and } \forall c_a, c_b \in C : c_a \neq c_b \Rightarrow c_a \cap c_b = \emptyset. \qquad (1)$$

Note that in (1) we do not expect clustering $C$ to include all the items in $O$. This is firstly due to the fact that clusterings can be performed on non-overlapping subsets of the data and secondly that data can contain outlier items which are not possible to include in a cluster.

Our conceptual framework is realized in the visual environment named CIVA (Interactive Visual Cluster Analysis). CIVA incorporates different types of conventional visual analysis views: histograms, scatter plots, parallel coordinates, etc. for regular variables; functions graphs and animated scatter plots for temporal variables. Function graph displays all $g(o,t)$, with $x$-axis as time and $y$-axis as the function values. For a better overview of temporal variables, we implement an animated scatter plot where displayed points move while an internal time variable changes. All of these views are linked through a brushing mechanism, except the animated point graph. This brushing mechanism is similar to *composite brushing* proposed by Allen and Ward [Martin and Ward 1995]. The result of brushing applied on a view is $b \subseteq O$, which is then combined with existing brushes by the boolean operator $S$ being $S \in \{\cup, \cap, \neg\}$, where $\cup$ represents union, $\cap$ represents intersection and $\neg$ represents not operator. Every brush $b$ is combined with the previous $b_i$ by applying the $S$ operator: $b_{i+1} = S(b_i, b)$. In addition to the existing brushing scheme, we append a brushing operator on cluster level represented as $b = c \in C$. For simplicity, in the following we will denote the final set of brushed items as $b_L$. Moreover, brushes on temporal variables select a time interval in addition to a list of items. This enables the analyst to concentrate on different time intervals while doing the analysis.

To demonstrate our framework in the following sections, we employ the Iris dataset [Fisher 1936], which is extensively used in data mining literature. The dataset consists of 150 samples from the three species of Iris flowers (Iris Setosa, Iris Versicolour, and Iris Virginica), where four features were measured from each sample, here being our regular variables. The measured features are the sepal length ($f_1$), the sepal width ($f_2$), the petal length ($f_3$) and the petal width ($f_4$). All the figures up-to Section 5 are visualizations of this dataset.

### 3.1 Cluster tendency view

An inherent problem in cluster formation is an assessment of cluster tendencies. Transforming to our IVA viewpoint, the question would be whether the current selection $b_L$ contains any cluster tendencies. One of the cluster tendency evaluations is based on visualizing the similarities between items.

To visualize cluster tendencies, we propose *Cluster Tendency View* (tendency view), which is based on the dissimilarity matrix visualization approach presented by Bezdek and Hathaway [Bezdek and Hathaway 2002]. In the dissimilarity matrix, $M$, every element $m_{i,j}$ represents dissimilarity measure between the items $o_i$ and $o_j$, which is smaller for more similar items. Here, for every matrix element, we compute the sum of mutual distances between each pair of variables. Importantly, the matrix is computed not for every item, but only for $b_L$, for which we define dissimilarity matrix $M$ as follows:

$$m_{i,j} = \sum_{k=1}^{p} w_k * d\left(f_k(o_i), f_k(o_j)\right) + \sum_{k=p+1}^{q+p} w_k * \left[\frac{\sum_{t=t_0}^{t_1} d\left(g_k(o_i,t), g_k(o_j,t)\right)}{t_1 - t_0}\right], \qquad (2)$$

where $d(\cdot,\cdot)$ is a distance function, $w_k$ are weights that are specified by the user which can emphasize or suppress (zero) certain variables. Distance functions are essential elements of cluster analysis and there is a large number of distance functions proposed in the literature [Shi et al. 2009]. Especially with time dependent variables, distance function definition should consider domain specific criteria. In this paper, Euclidean distance is preferred for $d(\cdot,\cdot)$. However, our methods are not bound to a specific distance function and $d(\cdot,\cdot)$ should be chosen to fulfill domain specific constraints prior to analysis. Time dependent variables $g$ are computed within the given time interval $[t_0, t_1]$. This time interval is interactively determined by the logical combination of temporal brushes.

Importantly, on every change of the selection $b_L$ or a weight $w_i$, the dissimilarity matrix $M$ is automatically recomputed and visual-

ized in tendency view. This mechanism enables the tight integration of tendency view into linked view system. Once the matrix is computed, it is normalized and visualized in tendency view using a color transfer function.

Referring back to the Iris dataset, we assign the corresponding weights equally, i.e., $w_{f_1} = w_{f_2} = w_{f_3} = w_{f_4} = 1$. Resultant dissimilarity matrix is shown in Fig. 3 (left). If we like to retrieve the cluster tendency by emphasizing the contribution of variables $f_3$ and $f_4$, we can achieve this by increasing $w_{f_3}$ and $w_{f_4}$ compared to $w_{f_1}$ and $w_{f_2}$ (Fig. 3 (right)). In the dissimilarity matrix, similar items are rendered in saturated red colors, while dissimilar ones in pale blue colors. Moreover, the matrix is ordered according to Ward's classification procedure, by which we construct new row and column orderings by iterative minima retrieval [Ward Jr 1963]. Although, the choice of the ordering algorithm can change the resulting visualization here, they will provide similar results regarding the clustering tendency of the selection. Therefore, Ward's method is preferred as it is a widely used classification method in data mining literature.

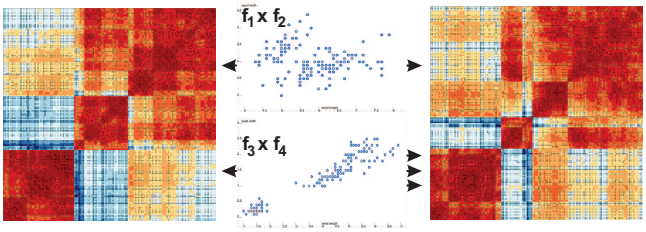



Figure 3: The dissimilarity matrix with the equal weights $w_{f_1} = w_{f_2} = w_{f_3} = w_{f_4} = 1$ (left). To emphasize the contribution of $f_3$ and $f_4$ variables, we tripled (tripled arrow) the weights of $f_3$ and $f_4$; i.e., $w_{f_3} = w_{f_4} = 3$ (right). For better understanding the corresponding scatterplots for both variable combinations are shown in the middle.

## 3.2 Integrated Clustering

Majority of the clustering methods operate on (dis)similarity matrices [Tan et al. 2006] and the expert has to decide on a suitable distance function with a set of dimensions to construct this matrix prior to performing clustering. Tendency view provides an interactive mechanism to construct, compare and evaluate the $M$ matrix (2) of the existing selection $b_L$. After discovering an appropriate $M$, user can continue with clustering $b_L$. In CIVA, we integrate clustering algorithms and manual grouping techniques operating on $M$. For instance, the user can apply k-means or hierarchical clustering [Tan et al. 2006] algorithms on $b_L$ using directly $M$ as a parameter to the algorithms. Here, we use a more robust implementation of k-means by utilizing a method called partitioning around medoids [Kaufman and Rousseeuw 2005]. One additional cluster formation solution we propose is to employ directly the tendency view to manually draw groups on the view's diagonal. Our intention here is not to replace clustering algorithms, but to allow for more precise cluster evaluation process. For instance, using the tendency view from Fig. 3 (left) we perform manual clustering where five groups are formed. We can clearly spot only cluster (5); while the upper clusters are not that clearly separable and requires further analysis (Fig. 4). All the clusterings, which are created manually, algorithmically or on different dimensions of the dataset become a part of the dataset itself at the end of this phase.

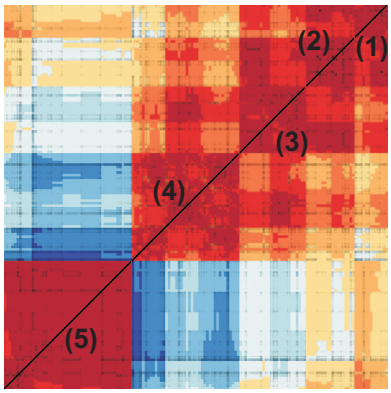



Figure 4: The interactive cluster formation using the tendency view and dissimilarity matrix. User draws a set of edges on the matrix diagonal, where each edge represents a cluster. This allows to create manual clusterings based on user priorities. Here, we specify five clusters that can be visually separated from each other.

## 3.3 Parallel Cluster View

Clusters provide high level information about the internal structure of the data. In order to analyze the underlying relations in the data, we incorporate clusters as data dimensions in IVA operations. Accordingly, we introduce *Parallel Cluster Views* (cluster view) to explicitly use clusters in brushing & linking operations. PCV, which is analogous to parallel sets proposed by Kosara et al. [Kosara et al. 2006], displays a number of different clusterings and enables the user to make selections at cluster level.

In a cluster view, vertical axes visualize different clusterings $C_k$, where $k$ indicates the order of the clustering axis, i.e., for the left-most axis, $k = 0$; and each curve between the axes represent a single data item, $o_i$. All axes contain a set of clusters where each cluster is represented by a different color. Curves between axis $k$ and $k+1$ are colored with respect to the colors of the clusters they are members of in $C_k$. This coloring schema improves the comprehension of membership changes between different clusterings.

Ordering of the items in a cluster is crucial on the perception of changes in membership relations. The ordering for the curves are computed by considering the overlapping members between $C_k$ and $C_{k+1}$. Firstly, the items are grouped into branches, where a branch represents the items $c_i \cap c_j$ where $c_i \in C_k, c_j \in C_{k+1}$. Secondly, the curves within a cluster are drawn regarding the cluster's branches with the next clustering. Using this methodology, we make sure that items that share the same cluster in $C_k$ and $C_{k+1}$ never overlap in between two neighboring clusterings.

Cluster view provides an intuitive mechanism to analyze and compare a number of different clusterings. For instance, Fig. 5a displays three clusterings ($C_{0,1,2}$) (formed using the weights $w_{f_1} = w_{f_2} = 1$ and $w_{f_3} = w_{f_4} = 0$, Fig. 4) which are: manual grouping ($C_0$) performed in Fig. 4, k-means clustering with $k$ parameter as 5 ($C_1$) and a hierarchical clustering ($C_2$). As hierarchical clustering contains a set of levels, the appropriate level to use in the visualization is determined interactively by the analyst.

Cluster view is tightly integrated into interactive analysis cycle by cluster level brushes and linkage to the selection mechanism in all other views. The user can select any number of clusters using one of $S$ operators. For instance, in Fig. 5b a cluster lever brush ($b_1$), highlights the items (arrows) in the other linked views in a focus+context manner. In the presented configuration of views, we

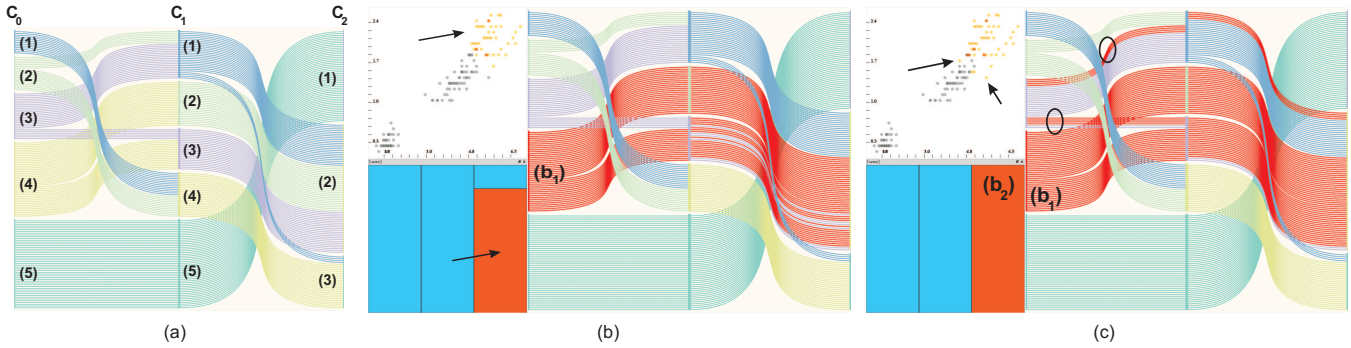



Figure 5: a) Parallel cluster view showing three clusterings ($C_{0,1,2}$), wherein $C_0$ and $C_1$ are composed of five clusters and $C_2$ of three clusters. Curves represent the item membership relations within the clusterings. Note that clusters (5),(5) and (1) in sequential clusterings from left to right are identical. b) Cluster brush ($b_1$) is used to select items representing members of cluster (4) in clustering ($C_0$). Brushed items are also highlighted in the accompanied scatter plot and the histogram that depicts a ratio of selected items per the specie. c) Combining selections on cluster and object level to analyze items of interest according to their cluster membership. Cluster brush $b_1$ and subsequent application of the $\cup$ operator with the histogram brush $b_2$ reveals how brushed items are distributed in all clusterings (circles).

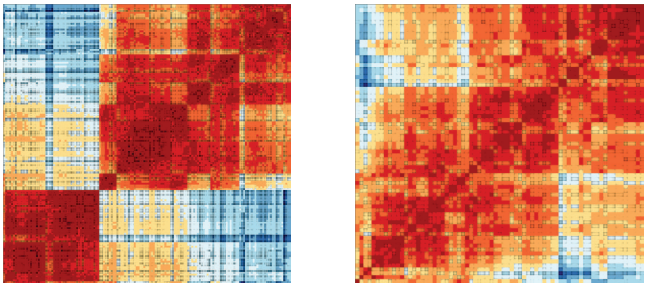
select cluster 4 and directly observe all the items, members of cluster 4, in the accompanying histogram and scatter plot. In the histogram, ratio of species in selected items can be observed.

Moreover, brushes in other views are linked to the cluster view and can be combined with cluster level brushes. In Fig. 5c the cluster level brush ($b_1$) is combined with histogram brush ($b_2$) with the $\cup$ operation, which selects additional items being members of the third specie.

## 4 Cluster Analysis Procedures

In the following, we describe cluster analysis procedures in cluster formation and cluster evaluation phases. Formation phase includes assessment of cluster tendency, while evaluation phase involves cluster comparison, coherence and stability. These procedures are showcased in CIVA environment.

Cluster formation phase starts with the exploration of a data subset and/or dimensions suitable for clustering. Tendency view assists the analyst to evaluate the clustering tendency of the selection before clustering. In Fig. 6a we see the tendency view of a selection which is suitable for clustering as it contains apparent structures in it. However in Fig. 6b, tendency view displays almost a random distribution which would yield to a less successful clustering. These

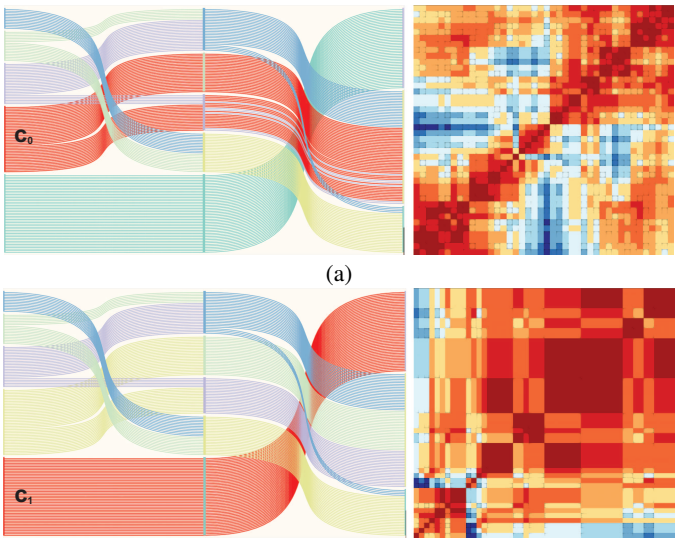


(a) (b)

Figure 6: Two tendency views depicting different clustering tendencies. (a) contains apparent structure suitable for clustering. (b) is not preferable for clustering as no structure is easily visible.

views provide the reasoning to favor a specific selection and/or dimension combination. Therefore, in this case (a) is preferred for clustering.

In the evaluation phase, clusters are visualized and compared to assess their meaning and validity . Evaluation of a cluster begins with *coherence* assessment. Tight cooperation between tendency view and cluster view in assessing cluster coherence can be seen in Fig. 7 where two cluster level brushes (to select clusters $c_0$ and $c_1$) are visualized with their corresponding dissimilarity matrices. Selected cluster $c_0$ results in a tendency view which depicts sub-




(a)

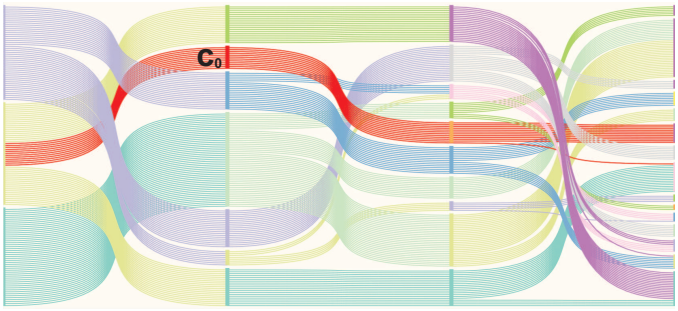
(b)

Figure 7: Two clusters ($c_{0,1}$) and their corresponding cluster views. As $c_0$ contains apparent structures, it requires further refinement. $c_1$ results in a uniform tendency view and it can be regarded as a satisfactory cluster.

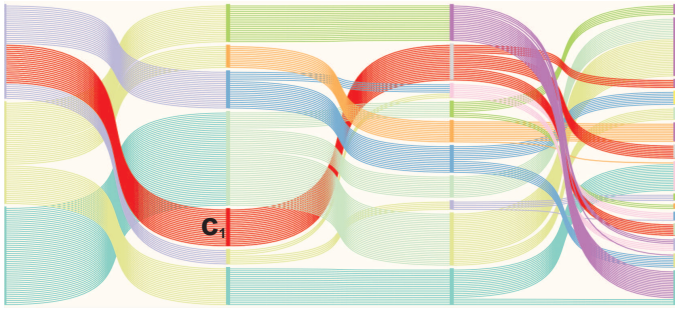
groups, which means it needs further refinement to get more valid results. However, tendency in the bottom row indicates that distances between items in $c_1$ are uniform, meaning that the cluster can be considered as a valid cluster regarding inter-cluster distances (i.e., coherence). The analyst can declare $c_1$ as satisfactory and con-

centrate on finding better clusterings for the items in $c_0$.

In cluster analysis, evaluating *cluster stability* is a critical technique to validate clusters. In order to demonstrate cluster stability evaluation, we made several clusterings of the same selection using different $k$ parameters ($k = 3, 7, 10, 15$) for the k-means algorithm. If the items in a certain cluster tend to stay in the same cluster as $k$ increases, this cluster can be considered as stable. Cluster labeled $c_0$ in Fig. 8 is a stable cluster as most of its members stay in the same cluster in all the clusterings. On the other hand, members of




(a)




(b)

Figure 8: Using cluster views to evaluate cluster stability. Cluster view visualize four clusterings made with k-means algorithm ($k = 3, 7, 10, 15$). Cluster $c_0$ is a stable cluster as most of its members are in the same cluster in consecutive clusterings. However $c_1$ needs further refinement as its members are spread across different clusters in later results

$c_1$ are spread among a number of different clusters with $k = 15$. Compared to $c_0$, $c_1$ can be regarded as an unsatisfactory cluster as there is an inconsistency between different runs of the clustering algorithm.

## 5 Case study: cluster analysis of mixed lipid bilayers

To demonstrate the usefulness of our approach, we present a study of cluster formations in lipid bilayers of biological membranes. Here, we do not focus on the cluster formation phase, but rather to showcase the importance of the evaluation phase. Our case study proved to be beneficial in getting better insight into data, which then led to new points of discussion on lipid bilayers. Biological membranes are active players in most biological processes and the dynamic behavior of the lipids which constitute them is decisive. In an attempt to understand biological membranes and membrane proteins, lipid bilayers have been and still are extensively studied. Molecular dynamics (MD) simulations are utilized as powerful tools to describe the atomic structure and dynamics of lipid bilayers, since detailed structural data of the most biologically relevant phases is difficult to obtain experimentally. In particular mixtures containing more than one lipid type have been studied in order to understand how different lipid types cluster together and can lead to inhomogeneity in biological membranes [Broemstrup and Reuter 2010]. Unfortunately, data generated by MD simulations can be rather tedious to analyze. Moreover, the analysis is nowadays performed on a non-interactive basis thus inactivating the user in the analysis of, e.g., cluster formation and cluster evaluation. In this study, we have found our framework to be highly beneficial for lipid clustering analysis of MD data.

The MD dataset of a mixed lipid bilayer [Broemstrup and Reuter 2010] is constituted of DMPC (dimirystoilphosphatidylcholine) and DMPG (dimirystoilphosphatidylglycerol) lipids. Each lipid is represented by one particle, localized at the position of the phosphorus atom. The particles (items) undergo stochastic oscillation movements in $x$ and $y$ directions, and only slight variations along the $z$ axis, where the number of time steps in our simulation equals 1640. We extend the dimensionality of the dataset by computing the first derivatives of the movements, $x'$ and $y'$. Additionally, the items have a regular variable representing their categories as mentioned above; i.e., DMPC and DMPG. The lipids are positioned in two separated layers along the $z$ axis and computational biologists are mostly interested in analyzing these lipid layers separately. Therefore, we limit our analysis to one of the $z$-layers by brushing the upper $z$-layer prior to the analysis.

To perform clustering on different time intervals, we brush the time intervals; $[50, 150], [700, 800]$ and $[1540, 1640]$. These intervals were clustered separately using hierarchical clustering based on $x$ and $y$ coordinates. We discovered groups of items, separated at the beginning, which form a cluster in the middle of the simulation and separate again at the end of the simulation. In Fig. 9 three animated scatter plots are shown, which depict the distribution of such a group ($c$). These plots display the same items in the $x-y$ plane

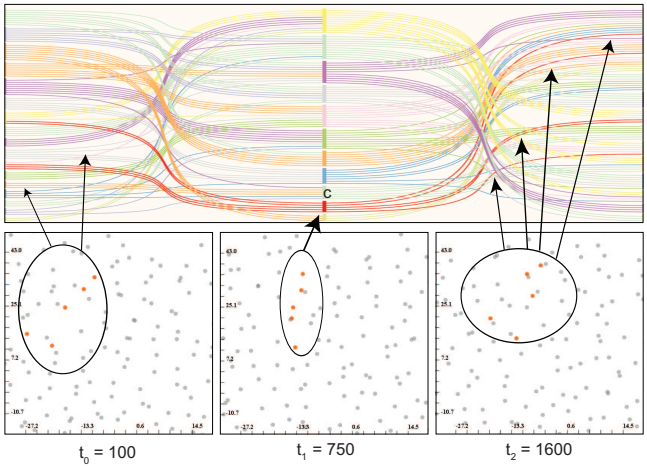



Figure 9: Using the cluster view to compare clusterings performed in three distinct time intervals, $[50, 150], [700, 800]$, and $[1540, 1640]$. The analysis reveals a number of groups that are separate at $t_0$, cluster together at $t_1$ and separate again at $t_2$. Animated scatter plots justify this claim.

at different time steps ($t_0 = 100, t_1 = 750, t_2 = 1600$). The scatter plots visualize how two groups of items at $t_0$, form a single group at $t_1$, and then separate into smaller groups again at $t_2$. This type of behavior is extremely important to follow, i.e., whereas groups stay together once they are formed, and is relevant for the formation of

lipid domains (rafts) in biological membranes. Moreover the ability of interactively following the composition of the clusters has no equivalent in the non interactive methods for MD analysis.

In general, computational biologists cluster MD data using only their $x$ and $y$ coordinates. However, in some cases it would be beneficial to form clusterings according to their velocities, $x'$ and $y'$, and evaluate the results. The same set of lipids is clustered using three different weight distributions $w = \langle w_x, w_y, w_{x'}, w_{y'} \rangle$. Clustering $C_0$ is formed with $w = \langle 1,1,0,0 \rangle$, $C_1$ with $w = \langle 1,1,1,1 \rangle$, and $C_2$ with $w = \langle 0,0,1,1 \rangle$. In fig. 10, the first clustering is based solely on

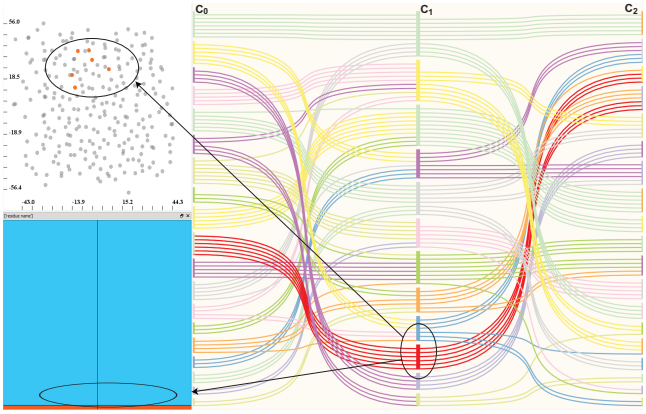



Figure 10: Using the cluster view to compare clusterings performed on different dimension subsets. $C_0$ is created on lipid positions, $C_2$ on lipid velocities and $C_1$ on a combination of all these dimensions. The selected cluster reveals a group of items which constitute a velocity cluster in addition to their positional cluster. Histogram displays lipid type distribution.

positions, the second one on a combination of positions and velocities, and the third one only on velocities. This provides a very clear image on how much the clusterings differ in their positions with respect to their velocities. We can easily evaluate whether there is a correlation between these two types of clusterings by cluster selections. Notably, an interesting outcome of this analysis is when an equal number of DMPC and DMPG type lipids form a positional cluster (from $C_0$), these items also have similar velocities. On the other side, positional clusters containing an unequal number of different type of lipids do not have the same velocities. Fig. 10 displays one of the clusters discovered in the analysis, by utilizing the cluster view.

In one clustering procedure of lipid bilayers, we employed only the last 100 time steps from all 1640 frames to perform clustering analysis, due to the system stabilization [Broemstrup and Reuter 2010]. The clustering analysis pointed out a new issue that is to evaluate the influence of 'jumpers' on the clustering process. After the visualization of the derivatives $x'$ and $y'$, we found out that nearly half of all the items exhibit big 'jumps' at a certain point of MD simulation. These jumpers correspond to atoms exiting at one side of the simulation bounding box and entering through the opposite one, which clamps the atoms along boundaries. Nevertheless, usually all the simulated particles are employed for the cluster analysis. On one hand, the exploited clustering technique [Broemstrup and Reuter 2010] that is used to analyze the MD simulation, takes every time step separately. Therefore such particles do not have to be considered as outliers. On the other hand, when clustering is performed over a time interval, where dissimilarity matrix summation is involved, these jumpers can cause cluster instability when jumping from one cluster to another. Once clustering is done, clusters are analyzed according to the distribution of their items' categories; i.e., DMPC or DMPG.

We perform a hierarchical clustering wherein the jumpers were included. Consequently we have the possibility to evaluate the effect of jumpers on the clustering by means of the cluster view and other linked views (Fig. 11). In the figure, the clusterings represent 5

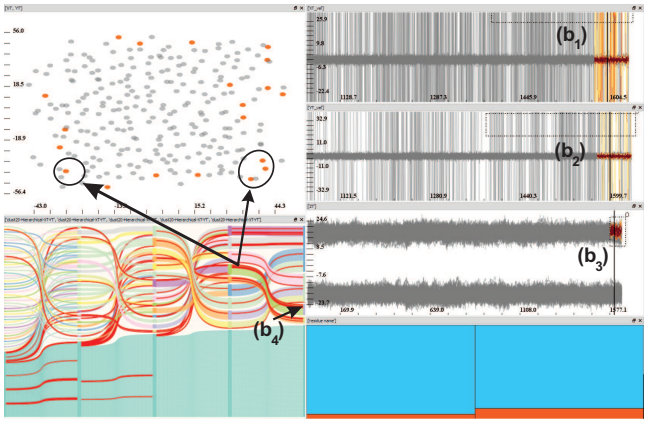



Figure 11: Discovery of jumpers within existing clusterings. The composed brush $b_L = (b_1 \cup b_2) \cap b_3$ selects the jumpers that are displayed in the animated scatter plot. The cluster view depicts their occurrence in individual clusterings. The cluster containing 4 jumpers was selected ($b_L \cap b_4$) to reveal their position (arrows and circles) within the cluster. The histogram depicts classification of the jumpers into DMPC and DMPG classes.

levels of a single hierarchical clustering (bottom left) performed on the last 100 time steps. The function graphs (top three right) are used to render derivatives $x'$ and $y'$, where we employed brush $b_L = (b_1 \cup b_2) \cap b_3$ to pick up the jumpers. The third function graph delimits the selection only to the upper layer for 100 time steps ($b_3$), where all 1640 time steps are visible. For the better understanding of the particles' movements we used an animated scatter plot (top left), where a user can directly observe the selected items in a focus+context style in the $x-y$ plane. The selection in the parallel cluster view reflects the occurrence of jumpers in individual hierarchical levels. The jumpers participate more and more on clusterings as we increase the cluster level, where the lowest cluster represents non-clustered items. In the animated scatter plot we discover the relation of jumpers per individual cluster by making $b_L \cap b_4$ operation in the cluster view, which displays the group of jumpers from the same cluster (arrows and circles, delimited by $b_4$). As one can clearly see, one of them has radically moved (left circle), while still being the member of the same cluster. Additionally, the histogram (bottom right) depicts the classification of the jumpers into DMPC and DMPG classes. It can be deduced from the example that jumpers are not a crucial choice for the smaller clusters; however when building bigger ones, their presence should be taken into account.

## 5.1 Implementation Details

In order to maintain the interactivity of the proposed system, we are using GPGPU techniques to perform the operations that are computationally expensive. In our framework, the data items and the selections are passed to GPU as texture maps. The coloring (focus+context visualization) and selection for all the views is done on GPU using these textures. One costly operation in our framework is the calculation of the dissimilarity matrix (Equation 2) in

tendency view. Therefore, the computation of these values is done in the GPU using CUDA. The resulting values are then transferred to the visualization pipeline through the GPU memory. This mechanism ensures that the framework operates at interactive rates.

## 6 Conclusion

In this paper, we introduce a novel concept for visual cluster analysis, which tightly integrates cluster formation and cluster evaluation, embedded within interactive visual analysis. In cluster formation we assist the users to explore subsets of the data that are suitable for clustering. In cluster evaluation, we utilize techniques from data mining to assess the cluster validity. These techniques are based on cluster comparison, cluster tendency, cluster coherence, and cluster stability, and they prove to be beneficial in order to achieve a successful cluster analysis in IVA. Consequently, the integration of these techniques leads to a better understanding of the underlying relations in the data.

The realization of our framework, CIVA, enriches the conventional interactive visual tools with two specific views capable of integrating clusters into IVA. The cluster tendency view enables the evaluation of the current selection of items for possible clusterings. Moreover, it allows the assessment of the cluster coherence in existing clusterings. The parallel cluster view provides a visualization of the item-to-cluster relationship, the evaluation of cluster stability and coherence, and importantly, the cluster comparison. The cluster view selection mechanism allows to link cluster level selections with selections made in conventional views.

We demonstrated CIVA in molecular dynamics simulations analysis, where the presented techniques leads to new considerations in the discussion on lipid bilayers. We performed three cluster analyses, where we studied the influence of velocities, time intervals, and 'jumpers' on the simulation and its analysis.

As a future work, we will extend the scalability of the proposed views. Accordingly, in order to display few thousands of items in the tendency view, it is required to adapt the dissimilarity matrix appropriately. This can be achieved by displaying only a certain hierarchical level of dissimilarity matrix instead of the full resolution rows.

To conclude, with the proposed integration we managed to overcome the challenges that relate to the black-box nature of clustering algorithms. Moreover, we believe that our framework provides more reliable clusterings and when integrated into IVA, these clusterings provide even better insight into the underlying relations in the data.

## 7 Acknowledgements

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

BEZDEK, J., AND HATHAWAY, R. 2002. Vat: a tool for visual assessment of (cluster) tendency. In *Neural Networks, 2002.*, vol. 3, 2225 –2230.

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

FISHER, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7*, 7, 179–188.

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

JAIN, A. 2010. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters 31*, 8, 651–666.

KAUFMAN, L., AND ROUSSEEUW, P. 2005. *Finding Groups in Data: An Introduction to Cluster Analysis. Wiley's Series in Probability and Statistics*. John Wiley and Sons, New York.

KEIM, D., ANDRIENKO, G., FEKETE, J., GÖRG, C., KOHLHAMMER, J., AND MELANÇON, G. 2008. Visual analytics: Definition, process, and challenges. *Information Visualization*, 154–175.

KONYHA, Z., MATKOVIC, K., GRACANIN, D., JELOVIC, M., AND HAUSER, H. 2006. Interactive visual analysis of families of function graphs. *IEEE transactions on Visualization and Computer Graphics 12*, 6, 1373–1385.

KOSARA, R., BENDIX, F., AND HAUSER, H. 2006. Parallel sets: interactive exploration and visual analysis of categorical data. *Visualization and Computer Graphics, IEEE Transactions on 12*, 4, 558 –568.

LANGE, T., ROTH, V., BRAUN, M., AND BUHMANN, J. 2004. Stability-based validation of clustering solutions. *Neural Computation 16*, 6, 1299–1323.

LEX, A., STREIT, M., PARTL, C., KASHOFER, K., AND SCHMALSTIEG, D. 2010. Comparative analysis of multidimensional, quantitative data. *Visualization and Computer Graphics, IEEE Transactions on 16*, 6, 1027 –1035.

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

RINZIVILLO, S., PEDRESCHI, D., NANNI, M., GIANNOTTI, F., ANDRIENKO, N., AND ANDRIENKO, G. 2008. Visually driven analysis of movement data by progressive clustering. *Information Visualization 7*, 3, 225–239.

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., AND HAMANN, B. 2010. Integrating data clustering and visualization for the analysis of 3d gene expression data. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on 7*, 1, 64 –79.

SCHRECK, T., BERNARD, J., TEKUSOVA, T., AND KOHLHAMMER, J. 2008. Visual cluster analysis of trajectory data with interactive Kohonen Maps. In *IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST'08*, 3–10.

SEO, J., AND SHNEIDERMAN, B. 2002. Interactively exploring hierarchical clustering results. *IEEE Computer 35*, 7, 80–86.

SHARKO, J., GRINSTEIN, G., MARX, K., ZHOU, J., CHENG, C.-H., ODELBERG, S., AND SIMON, H.-G. 2007. Heat map visualizations allow comparison of multiple clustering results and evaluation of dataset quality: Application to microarray data. In *Information Visualization, 2007. IV '07. 11th International Conference*, 521 –526.

SHARKO, J., GRINSTEIN, G., AND MARX, K. 2008. Vectorized radviz and its application to multiple cluster datasets. *IEEE transactions on Visualization and Computer Graphics*, 1444–1427.

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

SIIRTOLA, H. 2004. Interactive cluster analysis. In *IV '04: Proceedings of the Information Visualisation, Eighth International Conference*, IEEE Computer Society, Washington, DC, USA, 471–476.

SMITH, S., AND JAIN, A. 1984. Testing for uniformity in multidimensional data. *IEEE transactions on pattern analysis and machine intelligence 6*, 1, 73–81.

SPRENGER, T., BRUNELLA, R., AND GROSS, M. 2000. H-BLOB: a hierarchical visual clustering method using implicit surfaces. *Proceedings Visualization 2000.*, 61–68,.

TAN, P., STEINBACH, M., AND KUMAR, V. 2006. *Introduction to data mining*. Pearson Addison Wesley Boston.

WARD JR, J. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association 58*, 301, 236–244.