*w*CLUTO: A Web-Enabled Clustering Toolkit[1]

Matthew D. Rasmussen, Mukund S. Deshpande, George Karypis*, James Johnson, John A. Crow, and Ernest F. Retzel

Department of Computer Science and Engineering, University of Minnesota, 4–192 EE/CS, Minneapolis, Minnesota 55455

As structural and functional genomics efforts provide the biological community with ever-broadening sets of interrelated data, the need to explore such complex information for subtle relationships expands. We present *w*CLUTO, a Web-enabled version of the stand-alone application CLUTO, designed to apply clustering methods to genomic information. Its first application is focused on the clustering transcriptome data from microarrays. Data can be uploaded by the user into the clustering tool, a choice of several clustering methods can be made and configured, and data are presented to the user in a variety of visual formats, including a three-dimensional "mountain" view of the clusters. Parameters can be explored to rapidly examine a variety of clustering results, and the resulting clusters can be downloaded either for manipulation by other programs or to be saved in a format for publication.

Methods for monitoring genome-wide mRNA expression changes such as oligonucleotide chips (Fodor et al., 1993) and cDNA microarrays (Schena et al., 1995) allow for the rapid and inexpensive monitoring of the expression levels of a large number of genes at different time points for different conditions, tissues, and organisms. Knowing when and under what conditions a gene or a set of genes is expressed often provides strong clues as to their biological role and function. Clustering algorithms are exploratory tools for analyzing large data sets and have proved to be essential for data analysis and for gaining insight on various aspects of the genetic machinery.

Since the development of the microarray technologies, a wide range of existing clustering algorithms have been used, and novel new approaches have been developed for clustering gene expression data sets. The most effective traditional clustering algorithms are based on the agglomerative clustering methodology, *K*-means, and self-organizing maps.

There are a variety of commercial tools used in microarray analysis. Among these are GeneSpring (Silicon Genetics, Redwood City, CA; http://www.silicongenetics.com), SpotFire Decision Site (http://www.spotfire.com), the Rosetta Resolver package (Rosetta Biosoftware, Kirkland, WA; http://www.rosettabio.com), Expressionist (Genedata, San Francisco; http://www.genedata.com), and others. All of these packages have substantial licensing fees and a variety of fixed analyses. In the public arena are M. Eisen's Cluster and TreeView packages (http://rana. lbl.gov/EisenSoftware.htm), and GeneCluster (v2.0; http://www-genome.wi.mit.edu/cancer/software/genecluster2/gc2.html) developed by the Cancer Genomics groups at the Massachusetts Institute of Technology (Boston) is available for download. These packages include a subset of analyses.

Two packages have recently been released in the open source community to address stand-alone (workstation-based) analyses. The first, BioConductor (Dudoit et al., 2003; Dudoit and Fridlyand, 2003), is a rich, open source analysis package developed on the extensive *R* statistical computing package (http://www.r-project.org). The second, *TM4* (Saeed et al., 2003), is a strong competitor to the commercial packages mentioned above. *TM4* includes a MySQL-based microarray database management system and a user-friendly set of analytical and clustering tools and is written in Java. Finally, there are Web-based clustering services and tools such as the EPCLUST (http://ep.ebi.ac.uk/EP/EPCLUST/) from the European Bioinformatics Institute (Hinxton, Cambridge, UK), the BioArray Software Environment (Saal et al., 2002) from Lund University (Lund, Sweden), and GEPAS (http://gepas.bioinfo.cnio.es/) available at the Spanish National Cancer Center (Madrid). These sites allow users to upload their own data and provide a limited number of clustering algorithms. The Stanford Microarray Database (SMD; Sherlock et al., 2001) provides a set of Web-based clustering tools available to Stanford investigators and their collaborators based on Eisen's Cluster and TreeView packages.

In this paper, we describe *w*CLUTO, a Web-enabled application that we developed, available at http://cluto.ccgb.umn.edu/, that implements a vari-

[1] This work was supported in part by the National Science Foundation (award nos. ACI–0133464, CCR–9972519, EIA–9986042, ACI–9982274, DBI–0196197, DBI–9975806, DBI–9872565, DBI–0221524, and EIA–0224424), by the Army High Performance Computing Research Center (contract no. DAAH04–95–C–0008), by the U.S. Department of Agriculture (grant no. SCA 58–3625–8–117) funded by the North Central Soybean Research board and the United Soybean Board, and by the U.S. Department of Agriculture/CSREES/2002–35300–12621.

* Corresponding author; e-mail karypis@cs.umn.edu; fax 612–625–0572.

www.plantphysiol.org/cgi/doi/10.1104/pp.103.024885.

ety of clustering algorithms and allows the user to view the results using a number of different visualizations. The initial release of *w*CLUTO has been tailored to address the clustering and data analysis requirements of data sets obtained from gene expression studies. *w*CLUTO is built on our stand-alone, general-purpose clustering toolkit, CLUTO (http://www.cs.umn.edu/~cluto), a freely available software package for clustering low- and high-dimensional data sets and for analyzing the characteristics of the various clusters. To date, CLUTO has been successfully used to cluster data sets arising in many diverse areas including life sciences, information retrieval, customer relations marketing, and physical sciences. The entire set of programs required to implement *w*CLUTO on a different Web site can be downloaded from http://cluto.ccgb.umn.edu/download.

*w*CLUTO has been implemented as a set of Common Gateway Interface-based programs using a combination of Python, Perl, C, and C++ modules. *w*CLUTO allows the user to upload the data set(s) to cluster, to compute different clustering solutions using a variety of clustering algorithms, to visualize the results, and to compare the different clustering solutions, and then to download both the solutions and visualizations to the users' local computer.

SUPPORTED DATA FORMATS

A user can upload data to *w*CLUTO using two different formats. The first is a plain delimited ASCII file that contains the gene expression values to be clustered. *w*CLUTO supports most popular delimiters (tab, semicolon, comma, and space) and also allows the user to specify any additional delimiting characters. The user can select whether or not the first row and column of the uploaded file correspond to the names (i.e. labels) of the rows and columns, respectively.

In addition, microarray data are frequently stored in databases for exploration by other researchers. The SMD (Sherlock et al., 2001) is one significant example, as is The Institute for Genomic Research's TM4 (Saeed et al., 2003), and National Center for Genome Resources's Genex (http://www.ncgr.org/genex). SMD is significant because a variety of both plant and vertebrate data are stored at the Stanford site. Other instantiations of the database exist at the University of Minnesota. *w*CLUTO accepts data in the SMD.pcl format directly.

CLUSTERING ALGORITHMS

*w*CLUTO implements three different clustering algorithms that are based on the agglomerative, partitional, and graph-partitioning paradigms.

Agglomerative algorithms find the clusters by initially assigning each object to its own cluster and then repeatedly merging pairs of clusters until either the desired number of clusters has been obtained or all of the objects have been merged into a single cluster leading to a complete agglomerative tree. The key step in these algorithms is the method used to identify the pairs of clusters to be merged next. *w*CLUTO supports five different merging schemes. The first three are based on the classical single-link, complete-link, and group average approaches (Jain et al., 1999; Han et al., 2001], whereas the other two, called I2 and H2, are based on some recently introduced schemes that were motivated by research on partitional clustering (Zhao and Karypis, 2002). These schemes differ on how the similarity between the individual objects in the various clusters is combined to determine the similarity between the clusters themselves. The single-link criterion function measures the similarity of two clusters by the maximum similarity between any pair of objects from each cluster, whereas the complete-link criterion uses the minimum similarity. In general, both the single- and the complete-link approaches do not work very well because they either base their decisions on a limited amount of information (single link) or assume that all of the objects in the cluster are very similar to each other (complete link). On the other hand, the group average approach measures the similarity of two clusters by the average of the pair wise similarity of the objects from each cluster and does not suffer from the problems arising with single and complete link. The I2 and H2 schemes take an entirely different approach and treat the clustering process as an optimization problem by selecting the cluster pairs that optimize various aspects of the inter- and intracluster similarity of the resulting solution. The advantage of these schemes is that they lead to more natural clusters and agglomerative trees that are more balanced than the more traditional schemes. A precise description of these schemes is beyond the scope of this paper, and the reader should refer to Zhao and Karypis (2002) for a detailed description and comparative evaluation.

Partitional clustering algorithms find the clusters by partitioning the entire data set into a predetermined number of disjoint sets, each corresponding to a single cluster. This partitioning is achieved by treating the clustering process as an optimization procedure that tries to create high-quality clusters according to a particular function that reflects the underlying definition of the "goodness" of the clusters. This function is referred to as the clustering criterion function, and *w*CLUTO implements seven such criterion functions that measure various aspects of intracluster similarity, inter-cluster dissimilarity, and their combinations and has been shown to produce high-quality clusters in low- and high-dimensional data sets (Zhao and Karypis, 2003b). *w*CLUTO uses two different methods for computing the partitioning clustering solution. The first method computes a *k*-way clustering solution via a sequence of repeated bisections, whereas the second method computes the solution directly (in a fashion similar to traditional *K*-means-based algorithms). These methods are often referred to as repeated bisecting and direct *k*-way clustering, respectively. *w*CLUTO computes a direct *k*-way clustering as follows. Initially, a set of *k* objects is selected from the data sets to act as the seeds of the *k* clusters. Then, for each object, its similarity to these *k* seeds is computed, and it is assigned to the cluster corresponding to its most similar seed. This forms the initial *k*-way clustering. This clustering is then repeatedly refined so that it optimizes a desired clustering criterion function. This optimization is performed using a randomized incremental optimization algorithm that is greedy in nature, has low computational requirements, and produces high-quality solutions (Zhao and Karypis, 2003b). A *k*-way partitioning via repeated bisections is obtained by recursively applying the above algorithm to compute two-way clustering (i.e. bisections). Initially, the objects are partitioned into two clusters, then one of these clusters is selected and is further bisected, and so on. This process continues $k - 1$ times, leading to *k* clusters. Each of these bisections is performed so that the resulting two-way clustering solution optimizes a particular criterion function.

*w*CLUTO's graph-partitioning-based clustering algorithms use a sparse graph to model the affinity relations between the different objects and then discover the desired clusters by partitioning this graph (Karypis et al., 1999). To some extent, this approach is similar in spirit with that used by the partitional clustering algorithms described earlier; however, unlike those algorithms, the graph-partitioning-based approaches consider only the affinity relations between an object and a small number of its most similar other objects. As will be discussed later, this enables them to find clusters that have inherently different characteristics than those discovered by partitional methods. *w*CLUTO provides different methods for constructing this affinity graph and various post-processing schemes that are designed to help in identifying the natural clusters in the data set. The actual graph partitioning is computed using an efficient multilevel graph-partitioning algorithm (Karypis and Kumar, 1999) that leads to high-quality partitionings and clustering solutions.

Finally, an ideal clustering algorithm should employ methods that will allow it to automatically discover the natural clusters present in the data set. Unfortunately, there are no such techniques that are universally suitable for all kinds of data sets, and for this reason, all of *w*CLUTO's clustering algorithms require as input the number of clusters to be discovered. However, as will be discussed later, *w*CLUTO provides a number of visualizations that can help the user to semi-automatically determine the right number of clusters.

CHARACTERISTICS OF THE VARIOUS CLUSTERING ALGORITHMS

The various clustering algorithms provided by *w*CLUTO have been designed, and are well suited, for finding different types of clusters—allowing for different types of analyses. There are two general types of clusters that often arise in different application domains and different analysis requirements. What differentiates them is the similarity relations among the objects assigned to the various clusters. The first type contains clusters in which the similarity between all pairs of objects assigned to the same cluster will be high. On the other hand, the second type contains clusters in which the direct pair wise similarity between the various objects of the same cluster may be quite low, but within each cluster, there exist a sufficiently large number of other objects that eventually "connect" these low similarity objects. That is, if we consider the object-to-object similarity graph, then these objects will be connected by many paths that stay within the cluster and traverse high similarity edges. The names of these two cluster types have been inspired by this similarity-based view, and they are referred to as globular and transitive clusters, respectively. *w*CLUTO's partitional and agglomerative algorithms are able to find clusters that are primarily globular, whereas its graph partitioning and some of its agglomerative algorithms (e.g. single link) are capable of finding transitive clusters.

SIMILARITY MEASURES

*w*CLUTO's clustering algorithms treat the objects to be clustered as vectors in a multidimensional space and measure the degree of similarity between these objects using the cosine function, the Pearson's correlation coefficient, extended Jaccard coefficient (Strehl and Ghosh, 2000), or a similarity measure derived from the Euclidean distance of these vectors. The first two similarity measures can be used by all clustering algorithms, whereas the last two can be used only by the graph-partitioning-based algorithms.

By using the cosine and correlation coefficient measures, two objects are similar if their corresponding vectors point in the same direction (i.e. they have roughly the same set of features and in the same proportion), regardless of their actual length. In the case of Pearson's correlation coefficient, the vectors are obtained by first subtracting their average value. On the other hand, the Euclidean distance does take into account both direction and magnitude. Finally, similarity based on extended Jaccard coefficient accounts both for angle as well as magnitude. These are some of the most widely used measures, and have been used extensively to cluster gene expression data sets (Zhao and Karypis, 2003a).

COMPUTATIONAL REQUIREMENTS

*w*CLUTO's algorithms have been optimized for operating on very large data sets both in terms of the number of objects as well as the number of dimensions. Nevertheless, the various clustering algorithms have different memory and computational scalability characteristics. The agglomerative-based schemes can cluster data sets containing 2,000 to 5,000 objects in under a minute, but due to their memory requirements, they should not be used to cluster data sets with more than 10,000 objects. The partitional algorithms are very scalable both in terms of memory and computational complexity and can cluster data sets containing several tens of thousands of objects in a few minutes. Finally, the complexity of the graph-based schemes is usually between that of agglomerative and partitional methods and maintains the low memory requirements of partitional schemes.

VISUALIZATIONS

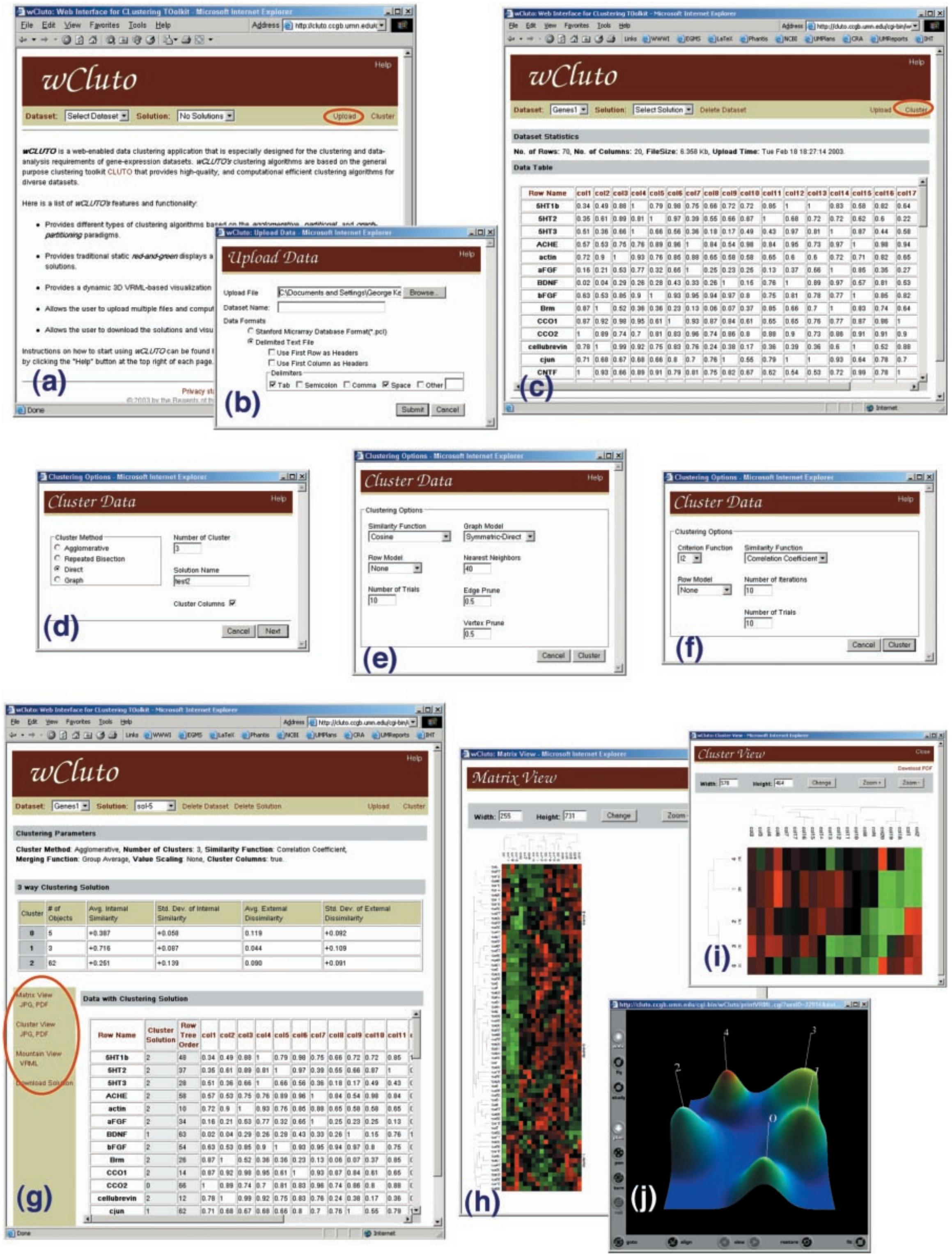
*w*CLUTO can produce three different visualizations of the clustering solutions (illustrated in Fig. 1). The first, called matrix view, is the traditional red-and-green intensity rendering of the data set in terms of a matrix whose rows correspond to the various objects (e.g. genes) and whose columns correspond to the various variables that describe each object (e.g. conditions). The matrix view conveys information about the underlying clustering solution by re-ordering the rows (and possibly columns) of this matrix according to a hierarchical clustering of the rows (and columns). This re-ordering is performed so that the resulting one-dimensional view of the data puts in successive positions similar sub-trees. The second, called cluster view, is similar to the matrix view, but instead of viewing the individual rows, it displays the various cluster centers and is well suited for visualizing clustering solutions for large number of clusters. The third, called mountain view, is a dynamic three-dimensional Virtual Reality Modeling Language-based visualization of the various clusters that are being projected on the plane so that they preserve as much as possible the inter-cluster similarities or distances. This visualization is obtained by projecting the clusters on a two-dimensional plane using multidimensional scaling (Duda et al., 2001), and then representing each cluster by a mountain whose height, total volume, and color intensity and shading encodes various aspects of its size and coherence. The matrix and cluster views are displayed using either JPG-images or PDF files, whereas the mountain view requires the user to first install a Virtual Reality Modeling Language plugin before viewing it and manipulating the image.

One of the key features of *w*CLUTO's matrix view is that, irrespective of the method used to obtain the clustering solution, it produces a hierarchical clustering of the objects (i.e. rows). In the case of agglomerative clustering, this hierarchical clustering corresponds to the solution computed by the agglomerative algorithm; however, in the case of the partitional and graph-partitioning algorithms, it computes a hierarchical tree that preserves the clustering solution that was computed. In this hierarchical solution, the objects of each cluster form a subtree, and the different sub-trees are merged to get an all-inclusive cluster at the end. These individual trees are combined in a meaningful way to accurately represent the similarities within each tree. This feature allows *w*CLUTO to produce hierarchical solutions even for very large data sets for which agglomerative algorithms are impractical due to their high computational and memory requirements.

These visualizations can be used to gain valuable information about the characteristics of each cluster and how the different clusters relate to each other. Specifically, the matrix and cluster views can be used to identify the variables on which the various objects in each cluster agree and to see how these variables change across the different clusters. Such visualizations are especially effective in allowing the user to easily identify the natural clusters in the data set and to determine the set of genes that show similar expression patterns. Additional information about the characteristics of each cluster can be obtained by looking at the mountain view visualization. For example, by looking at the height and color intensity of each mountain, a user can gain information about the strength of the pair wise similarity relations among the objects of each cluster and their cohesiveness. Similarly, the two-dimensional rendition of the cluster-to-cluster proximity relations provides additional information on how the various clusters are related beyond that provided by the one-dimensional ordering shown in the matrix and cluster views.

SESSION MANAGEMENT

*w*CLUTO allows users to upload multiple data sets and to compute multiple solutions for each of the different data sets that they have been uploaded. Once the user has finished analyzing their data sets, clustering solutions and visualizations can be downloaded. The user can selectively remove a clustering solution or can remove the complete data set. To ensure that the user can view all of the data sets and solutions while browsing, it is essential that we provide basic session management capabilities. Session management allows the Web server to associate the uploaded data sets and clustering solutions with a particular user. To provide maximum ease of use and convenience to the user, *w*CLUTO does not require user registration or the creation of Web accounts. To maintain maximum user anonymity, *w*CLUTO does



wCluto: Web Interface for CLustering TOolkit - Microsoft Internet Explorer

wCluto

Dataset: Select Dataset Solution: No Solutions Upload Cluster

wCLUTO is a web-enabled data clustering application that is especially designed for the clustering and data-analysis requirements of gene-expression datasets. *wCLUTO's* clustering algorithms are based on the general purpose clustering toolkit CLUTO that provides high-quality, and computational efficient clustering algorithms for diverse datasets.

Here is a list of *wCLUTO's* features and functionality:

- Provides different types of clustering algorithms based on the *agglomerative*, *partitional*, and *graph-partitioning* paradigms.
- Provides traditional static *red-and-green* displays a
solutions.
- Provides a dynamic 3D VRML-based visualization
- Allows the user to upload multiple files and comput
- Allows the user to download the solutions and visu

Instructions on how to start using *wCLUTO* can be found
by clicking the "Help" button at the top right of each page.

(a)

wCluto: Upload Data - Microsoft Internet Explorer

Upload Data

Upload File: C:\Documents and Settings\George Ka Browse...

Dataset Name:

Data Formats
- ○ Stanford Micrarray Database Format(*.pcl)
- ● Delimited Text File
  - ☐ Use First Row as Headers
  - ☐ Use First Column as Headers

Delimiters: ☑ Tab ☐ Semicolon ☐ Comma ☑ Space ☐ Other

Submit Cancel

(b)

wCluto: Web Interface for CLustering TOolkit - Microsoft Internet Explorer

Dataset: Genes1 Solution: Select Solution Delete Dataset Upload Cluster

Dataset Statistics

No. of Rows: 70, **No. of Columns:** 20, **FileSize:** 6.358 Kb, **Upload Time:** Tue Feb 18 18:27:14 2003

Data Table

Row Name	col1	col2	col3	col4	col5	col6	col7	col8	col9	col10	col11	col12	col13	col14	col15	col16	col17
5HT1b	0.34	0.49	0.88	1	0.79	0.98	0.75	0.66	0.72	0.72	0.85	1	1	0.83	0.58	0.82	0.64
5HT2	0.35	0.61	0.89	0.81	1	0.97	0.39	0.55	0.66	0.87	1	0.68	0.72	0.72	0.62	0.6	0.22
5HT3	0.51	0.36	0.66	1	0.66	0.56	0.36	0.18	0.17	0.49	0.43	0.97	0.81	1	0.87	0.44	0.58
ACHE	0.57	0.53	0.75	0.76	0.89	0.96	1	0.84	0.54	0.98	0.84	0.95	0.73	0.97	1	0.98	0.94
actin	0.72	0.9	1	0.93	0.76	0.85	0.88	0.65	0.58	0.58	0.65	0.6	0.6	0.72	0.71	0.82	0.65
aFGF	0.16	0.21	0.53	0.77	0.32	0.66	1	0.25	0.23	0.25	0.13	0.37	0.66	1	0.86	0.35	0.27
BDNF	0.02	0.04	0.29	0.26	0.28	0.43	0.33	0.26	1	0.15	0.76	1	0.89	0.97	0.57	0.81	0.53
bFGF	0.63	0.53	0.85	0.9	1	0.93	0.95	0.94	0.97	0.8	0.75	0.81	0.78	0.77	1	0.85	0.82
Brm	0.87	1	0.52	0.36	0.36	0.23	0.13	0.06	0.07	0.37	0.85	0.66	0.7	1	0.83	0.74	0.64
CCO1	0.87	0.92	0.98	0.95	0.61	1	0.93	0.87	0.84	0.61	0.65	0.65	0.76	0.77	0.87	0.86	1
CCO2	1	0.89	0.74	0.7	0.81	0.83	0.96	0.74	0.86	0.8	0.88	0.9	0.73	0.86	0.91	0.91	0.9
cellubrevin	0.78	1	0.99	0.92	0.75	0.83	0.76	0.24	0.38	0.17	0.36	0.39	0.36	0.6	1	0.52	0.88
cjun	0.71	0.68	0.67	0.68	0.66	0.8	0.7	0.76	1	0.55	0.79	1	1	0.93	0.64	0.78	0.7
CNTF	1	0.93	0.66	0.89	0.91	0.79	0.81	0.75	0.82	0.67	0.62	0.54	0.53	0.72	0.99	0.78	1

(c)

Clustering Options - Microsoft Internet Explorer

Cluster Data

Cluster Method
- ○ Agglomerative
- ○ Repeated Bisection
- ● Direct
- ○ Graph

Number of Cluster: 3

Solution Name: test2

Cluster Columns ☑

Cancel Next

(d)

Clustering Options - Microsoft Internet Explorer

Cluster Data

Clustering Options

Similarity Function: Cosine
Graph Model: Symmetric-Direct
Row Model: None
Nearest Neighbors: 40
Number of Trials: 10
Edge Prune: 0.5
Vertex Prune: 0.5

Cancel Cluster

(e)

Clustering Options - Microsoft Internet Explorer

Cluster Data

Clustering Options

Criterion Function: I2
Similarity Function: Correlation Coefficient
Row Model: None
Number of Iterations: 10
Number of Trials: 10

Cancel Cluster

(f)

wCluto: Web Interface for CLustering TOolkit - Microsoft Internet Explorer

Dataset: Genes1 Solution: sol-5 Delete Dataset Delete Solution Upload Cluster

Clustering Parameters

Cluster Method: Agglomerative, **Number of Clusters**: 3, **Similarity Function**: Correlation Coefficient, **Merging Function**: Group Average, **Value Scaling**: None, **Cluster Columns**: true.

3 way Clustering Solution

Cluster	# of Objects	Avg. Internal Similarity	Std. Dev. of Internal Similarity	Avg. External Dissimilarity	Std. Dev. of External Dissimilarity
0	5	+0.387	+0.058	0.119	+0.092
1	3	+0.716	+0.087	0.044	+0.109
2	62	+0.251	+0.139	0.090	+0.091

Matrix View
JPG, PDF

Cluster View
JPG, PDF

Mountain View
VRML

Download Solution

Data with Clustering Solution

Row Name	Cluster Solution	Row Tree Order	col1	col2	col3	col4	col5	col6	col7	col8	col9	col10	col11
5HT1b	2	48	0.34	0.49	0.88	1	0.79	0.98	0.75	0.66	0.72	0.72	0.85
5HT2	2	37	0.35	0.61	0.89	0.81	1	0.97	0.39	0.55	0.66	0.87	1
5HT3	2	28	0.51	0.36	0.66	1	0.66	0.56	0.36	0.18	0.17	0.49	0.43
ACHE	2	58	0.57	0.53	0.75	0.76	0.89	0.96	1	0.84	0.54	0.98	0.84
actin	2	10	0.72	0.9	1	0.93	0.76	0.85	0.88	0.65	0.58	0.58	0.65
aFGF	2	34	0.16	0.21	0.53	0.77	0.32	0.66	1	0.25	0.23	0.25	0.13
BDNF	1	63	0.02	0.04	0.29	0.26	0.28	0.43	0.33	0.26	1	0.15	0.76
bFGF	2	54	0.63	0.53	0.85	0.9	1	0.93	0.95	0.94	0.97	0.8	0.75
Brm	2	26	0.87	1	0.52	0.36	0.36	0.23	0.13	0.06	0.07	0.37	0.85
CCO1	2	14	0.87	0.92	0.98	0.95	0.61	1	0.93	0.87	0.84	0.61	0.65
CCO2	0	66	1	0.89	0.74	0.7	0.81	0.83	0.96	0.74	0.86	0.8	0.88
cellubrevin	2	12	0.78	1	0.99	0.92	0.75	0.83	0.76	0.24	0.38	0.17	0.36
cjun	1	62	0.71	0.68	0.67	0.68	0.66	0.8	0.7	0.76	1	0.55	0.79

(g)

wCluto: Matrix View - Microsoft Internet Explorer

Matrix View

Width: 255 Height: 731 Change Zoom

(h)

Cluster View

Width: 578 Height: 464 Change Zoom+ Zoom-

(i)

(j)

Figure 1. Examples of *w*CLUTO's interface and page views. a, Welcome screen. b, Data upload dialog window. c, Data set view page. d through f, Clustering dialog windows. g, Clustering view page. h through j, Visualization pages.

not even make use of cookies. Session management is done by embedding a unique string, known as session-ID, in the URL. The first time a user visits the home page of *w*CLUTO, a unique session-ID is assigned to that user; *w*CLUTO identifies the user based on the session-ID. All of the Web pages served to that user will have that session-ID embedded in the URL.

USING *W*CLUTO

The very first step that a user has to take to start using *w*CLUTO is to upload a data file. This is accomplished by clicking the "Upload" button on *w*CLUTO's main page (Fig. 1a). Once this is done, a pop-up window will appear (Fig. 1b) that asks the user to supply the name of the locally stored file that contains the data to be clustered. The actual data transfer occurs when the user clicks the "Submit" button. Once the file has been uploaded, *w*CLUTO displays some basic statistics on the particular data set and presents the user with a table view of the actual data (Fig. 1c). Using this table view, the user can view their data set and if desired, by clicking at the column name, sort the rows according to the values of the different columns.

Once a data set has been uploaded, the user can then proceed to cluster it. This is accomplished by clicking the "Cluster" button in *w*CLUTO's navigation panel (circled item in Fig. 1c). By doing so, another pop-up window appears (Fig. 1d) that allows the user to select the type of clustering algorithm to use, the number of clusters, and the name of the clustering solution. Once the user clicks the "Next" button, another pop-up window appears that, depending on the previously selected clustering method, displays a set of options that allow control of various aspects of the particular clustering scheme (Fig. 1, e and f). Among others, they include the type of merging scheme, criterion function, similarity function, and the graph model to be used for the graph-partitioning-based clustering method.

After selecting the desired clustering method and options and after the clustering has been computed, *w*CLUTO displays the clustering solution page (Fig. 1g), which contains three pieces of information. First is the brief list of parameters used to obtain the solution. Second are various internal cluster quality measures that include the average pair wise similarity between each object of each cluster and its SD and the average similarity between the objects of each cluster to the objects in the other clusters and their SD. Third, it displays the table view of the actual data that has been augmented to contain two additional columns (the second and third columns of the table). The first column is the cluster number that each particular object belongs to, whereas the second number is the order of each object in the hierarchical tree-induced ordering of the data set (this is the same ordering by which the rows are ordered in the matrix view visualization). Again, the user can sort the data in this table by clicking at the column labels and thus re-order the data set in a meaningful fashion.

The left panel of the clustering solution page (Fig. 1g, circled items) also contains links to the various visualizations and allows the user to download the results onto a local computer. The three types of visualizations that are produced by *w*CLUTO are illustrated in Figure 1, h through j, which shows the matrix view, cluster view, and mountains view, respectively. The user can automatically resize the displayed images and by clicking the button at the top right of each page can download a high-quality PDF version of them for printing and inclusion in publications.

FUTURE ENHANCEMENTS

Although presently focused on microarray data that have been normalized using the tools of choice, the intent is to broaden the target data types that can be clustered. A logical extension from microarray analysis is transcriptome data from Serial Analysis of Gene Expression experiments, as well as incorporating derived information for standard microarray chips (e.g. protein families and metabolic reconstructions of known data) as clustering dimensions.

Another direction in which we are applying CLUTO technology is the incorporation of its core library in the data exploration tool Table View (Johnson et al., 2003). Table View is a Web-aware data exploration and visualization application, available to the community via Java WebStart, and provides coordinated multiple views of general data sets and clustering of the data. A user can use these different views to focus on and select data of interest and then make use of the embedded CLUTO library to create clusters for further exploration.

We have also been experimenting with the incorporation of the Table View-enabled CLUTO into the *TM4* (Saeed et al., 2003) open source microarray database and analysis system, using a server-based Java WebStart version of the *TM4* code base.

Received April 4, 2003; returned for revision April 27, 2003; accepted June 30, 2003.

LITERATURE CITED

Duda RO, Hart PE, Stork DG (2001) Pattern Classification. John Wiley & Sons, New York

Dudoit S, Fridlyand J (2003) Bagging to improve the accuracy of a clustering procedure. Bioinformatics **19:** 1090–1099

Dudoit S, Gentleman RC, Quackenbush J (2003) Open source software for the analysis of microarray data. Biotechniques Suppl. 45–51

Fodor SP, Rava RP, Huang XC, Pease AC, Holmes CP, Adams CL (1993) Multiplexed biochemical assays with biological chips. Nature **364:** 555–556

Han J, Kamber M, Tung AKH (2001) Spatial clustering methods in data mining: a survey. *In* H Miller, J Han, eds, Geographic Data Mining and Knowledge Discovery. Taylor and Francis, London, pp 188–217

Jain AK, Murty MN, Flynn PJ (1999) Data clustering: review. ACM Comput Surveys **31:** 264–323

Johnson JE, Stromvik M, Silverstein KAT, Crow JA, Shoop E, Retzel EF (2003) Tableview: portable genomic data visualization. Bioinformatics **19:** 1292–1293

Karypis G, Han EH, Kumar V (1999) Chameleon: a hierarchical clustering algorithm using dynamic modeling. IEEE Comput **32:** 68–75

Karypis G, Kumar V (1999) A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM J Sci Comput **20:** 359–392

Saal LH, Troein C, Vallon-Christersson J, Gruvberger S, Borg A, Peterson C (2002) BioArray Software Environment: a platform for comprehensive management and analysis of microarray data. Genome Biol **3:** 0.0003.1–0.0003.6

Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N, Braisted J, Klapa M, Currier T, Thiagarajan M et al. (2003) TM4: a free, open-source system for microarray data management and analysis. BioTechniques **34:** 374–378

Schena M, Shalon D, Davis RW, Brown PO (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science **270:** 467–470

Sherlock G, Hernandez-Boussard T, Kasarskis A, Binkley G, Matese J, Dwight S, Kaloper M, Weng S, Jin H, Ball C et al. (2001) The Stanford Microarray Database. Nucleic Acids Res **29:** 152–155

Strehl A, Ghosh J (2000) Value-based customer grouping from large retail data-sets. *In* BV Dasarathy, ed, SPIE Conference on Data Mining and Knowledge Discovery, Vol 4057. Bellingham, WA, pp 33–42

Zhao Y, Karypis G (2002) Evaluation of hierarchical clustering algorithms for document datasets. *In* K Kalpakis, N Goharian, and D Grossman, eds, Proceedings of the International Conference on Information and Knowledge Management. New York, pp 515–524

Zhao Y, Karypis G (2003a) Clustering in the life sciences. *In* M Brownstein, A Khodursky, eds, Functional Genomics: Methods and Protocols. Humana Press, Totowa, NJ

Zhao Y, Karypis G (2003b) Criterion Functions for Document Clustering: Experiments and Analysis. Machine Learning, Assinippi Park, Norwell, MA (in press)