wCLUTO: A Web-Enabled Clustering Toolkit

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

Department of Computer Science & Center for Computational Genomics and Bioinformatics
University of Minnesota, Minneapolis, MN 55455

CS Technical Report: 03–012

Abstract

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 wCLUTO*, 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 is 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 saved in a format for publication.*

1 Introduction

Methods for monitoring genome-wide mRNA expression changes such as oligonucleotide chips [5], and cDNA microarrays [16] 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 datasets, 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 datasets. The most effective traditional clustering algorithms are based either 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 [18], SpotFire Decision Site [20], the Rosetta Resolver package [14], Expressionist [7], as well as others. All of these packages have substantial licensing fees and a variety of fixed analyses. In the public arena are Eisen's Cluster and TreeView packages [3], and GeneCluster [1] developed by the Cancer Genomics groups in MIT are available for download. These packages include a subset of analyses. Further, there are specific packages written that attach to *R*, the popular open source statistics package. There are web-based clustering services and tools such as EPCLUST [4] from the European Bioinformatics Institute and GEPAS [19] available at the Spanish National Cancer Center. These sites allow users to upload their own data and provide a limited number of clustering algorithms. Finally, the Stanford Microarray Database [17] 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 wCLUTO, a web-enabled application we developed, available at http://cluto.ccgb.umn.edu/, that implements a variety of clustering algorithms and allows the user to view the results using a number of different visualizations. The initial release of wCLUTO has been tailored to address the clustering and data-analysis requirements of datasets obtained from gene-expression studies. wCLUTO is built on our stand-alone, general-purpose clustering toolkit, CLUTO [13], a freely available software package for clustering low- and high-dimensional datasets and for analyzing the characteristics of the various clusters. To date, CLUTO has been successfully used to cluster datasets arising in many diverse areas including life sciences, information retrieval, customer relations marketing, and physical sciences.

2 Methods

wCLUTO has been implemented as a set of CGI-based programs using a combination of Python, Perl, C, and C++ modules. wCLUTO allows the user to upload the dataset(s) to cluster, to compute different clustering solutions using a

variety of clustering algorithms, to visualize the results, to compare the different clustering solutions, and then download both the solutions and visualizations to the users' local computer.

Supported Data Formats A user can upload data to wCLUTO using two different formats. The first is a plain delimited ASCII file that contains the gene-expression values to be clustered. wCLUTO supports most popular delimiters (tab, semi-column, comma, space) and also allows the user to specify any additional delimiting characters. In addition, 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 is frequently stored in databases for exploration by other researchers. The Stanford Microarray Database (SMD) [17] is one significant example, as is TIGR's TM4 [15], and NCGR's Genex [6]. SMD is significant because a variety of both plant and vertebrate data is stored at the Stanford site. Other instantiations of the database exist at the University of Minnesota. wCLUTO accepts data in the SMD .pcl format directly.

Clustering Algorithms wCLUTO implements three different clustering algorithms that are based on the agglomerative, partitional, and graph-partitioning paradigms. These algorithms have been designed, and are well-suited, for finding different types of clusters—allowing for different types of analysis. wCLUTO's partitional and agglomerative algorithms are able to find clusters that are primarily globular, whereas its graph-partitioning and some of its agglomerative algorithms are capable of finding transitive clusters.

wCLUTO's agglomerative algorithms support the three traditional cluster merging schemes, namely group average, single-link, and complete-link [9, 8], as well as two other schemes that optimize various aspects of the inter- and intra-cluster similarity of the resulting clusters. A description of these new schemes is beyond the scope of this paper. These are evaluated and analyzed in detail in other publications [22].

wCLUTO provides two different partitional-based clustering algorithms that can be used to cluster the data into k user-specified clusters. The first method computes a k-way clustering solution via a sequence of repeated bisections, whereas the second methods 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. A key feature of wCLUTO's partitional clustering algorithms is that they treat the clustering problem as an optimization process that seeks to maximize or minimize a particular *clustering criterion function* defined globally over the entire clustering solution space. wCLUTO provides a total of seven different criterion functions that have been shown to produce high-quality clusters in low- and high-dimensional datasets [24]. In addition, these criterion functions are optimized using a randomized incremental optimization algorithm that is greedy in nature, has low computational requirements, and produces high-quality solutions [24].

wCLUTO'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 [11]. wCLUTO 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 dataset. The actual graph partitioning is computed using an efficient multilevel graph-partitioning algorithm [12] that leads to high-quality partitionings and clustering solutions.

wCLUTO's algorithms have been optimized for operating on very large datasets 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 datasets containing 2,000–5,000 objects in under a minute but due to their memory requirements they should not be used to cluster datasets with over 10,000 objects. The partitional algorithms are very scalable both in terms of memory and computational complexity, and can cluster datasets 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 maintain the low memory requirements of the partitional schemes.

Similarity Measures wCLUTO's clustering algorithms treat the objects to be clustered as vectors in a multi-dimensional space and measure the degree of similarity between these objects using either the cosine function, the Pearson's correlation coefficient, extended Jaccard coefficient [21], or a similarity measure derived from the Euclidean distance of these vectors. By using the cosine and correlation coefficient measures, two objects are similar if their corresponding vectors[1] 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. On the other hand, the Euclidean distance does take into account both direction and magnitude. Finally, similarity based on extended Jaccard coefficient account 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 datasets [23].

Visualizations wCLUTO can produce three different

[1] In the case of Pearson's correlation coefficient, the vectors are obtained by first subtracting their average value.

visualizations of the clustering solutions (illustrated in Figure 1). The first, called *matrix view* is the traditional *red-and-green* intensity rendering of the matrix whose rows (and possibly columns) have been re-ordered 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 subtrees. 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 3D VRML-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 obtain by projecting the clusters on a 2D plane using multidimensional scaling [2], 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 VRML 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 subtrees are merged to get an all inclusive cluster at the end. These individual trees are combined in a meaningful way, so that to accurately represent the similarities within each tree. This feature allows *w*CLUTO to produce hierarchical solutions even for very large datasets for which agglomerative algorithms are impractical due to their high computational and memory requirements.

Session Management *w*CLUTO allows users to upload multiple datasets, and to compute multiple solutions for each of the different datasets that they have currently uploaded. Once the user has finished analyzing their datasets, clustering solutions and visualizations can be downloaded. The user can selectively remove a clustering solution or can remove the complete dataset.

To ensure that the user can view all the datasets and solutions while browsing, it is essential we provide basic session-management capabilities. Session management allows the web-server to associate the uploaded datasets 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 creation of web-accounts. To maintain maximum user anonymity, *w*CLUTO does 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 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 the web-pages served to that user will have that session-ID embedded in the URL.

3 Using *w*CLUTO

The very first step that a user has to do in order to start using *w*CLUTO is to upload a data-file. This is accomplished by clicking at the "Upload" button on *w*CLUTO's main page (Figure 1(a)). Once this is done, a pop-up window will appear (Figure 1(b)) 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 dataset and presents the user with a table-view of the actual data (Figure 1(c)). Using this table-view, the user can view their dataset and if desired, by clicking at the column name, sort the rows according to the values of the different columns.

Once a dataset 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 Figure 1(c)). By doing so, another pop-up window appears (Figure 1(d)) that allows the user to select the type of clustering algorithm to use, the number of clusters, 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 (Figure 1(e,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 option and the clustering has been computed, *w*CLUTO displays the clustering solution page (Figure 1(g)) that 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 pairwise similarity between each object of each cluster and its standard deviation, and the average similarity between the objects of each cluster to the objects in the other clusters and their standard deviation. Third, it displays the table-view of the actual data that has been augmented to contain two additional columns (the second and third column 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 dataset (this is the same ordering

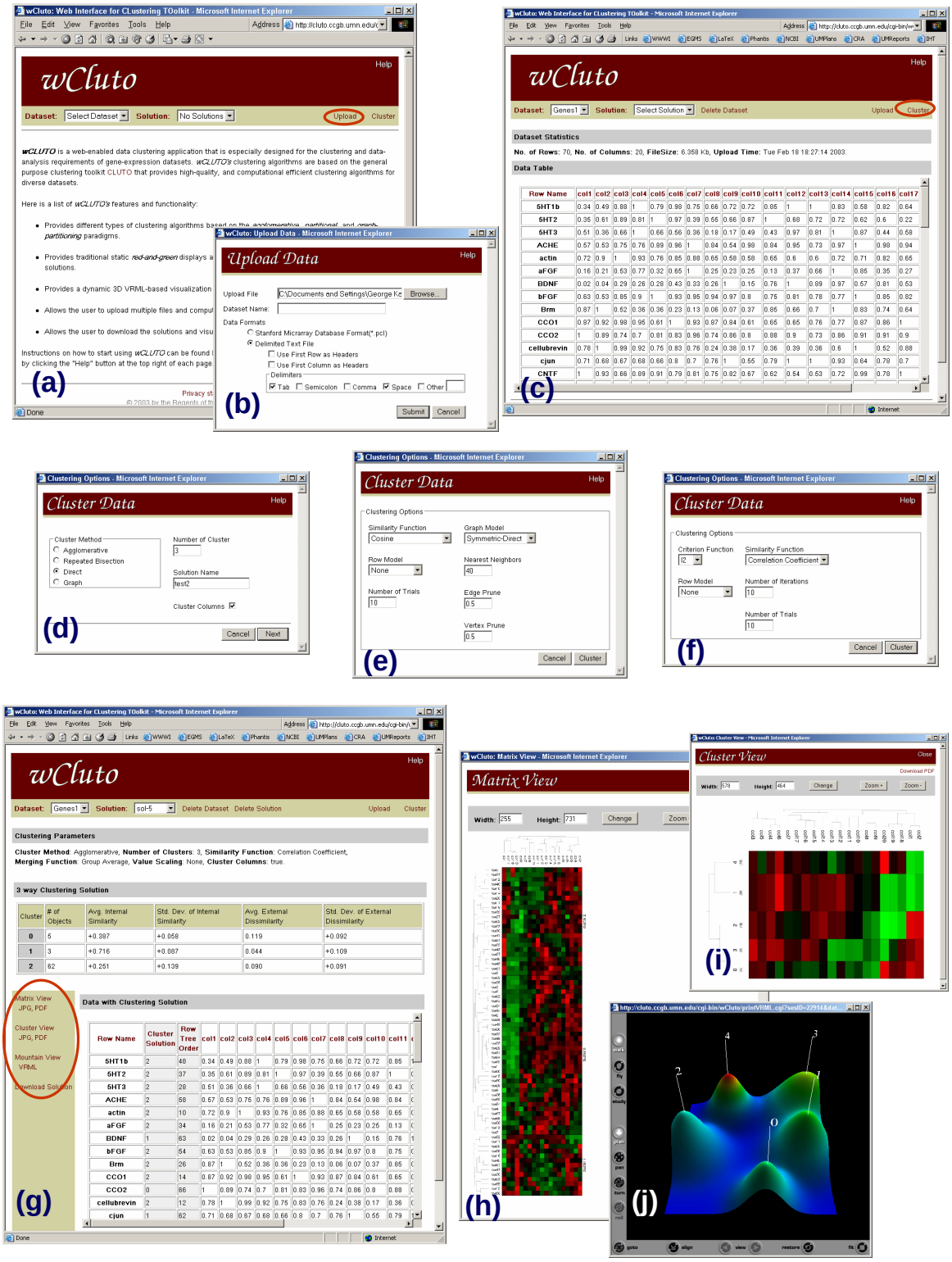

Figure 1: Examples of *w*CLUTO's interface and page-views. (a) Welcome screen. (b) Data upload dialog window. (c) Dataset view page. (d, e, f) Clustering dialog windows. (g) Clustering view page. (h,i,j) Visualization pages.

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 dataset in a meaningful fashion.

The left panel of the clustering solution page (circled items in Figure 1(g)) also contains links to the various visualizations and allow the user to download the results on to their local computer. The three types of visualizations that are produced by *w*CLUTO are illustrated in Figures 1(h,i,j), that show 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.

4 Future Enhancements

While presently focused on microarray data that has 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 SAGE 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 will be applying CLUTO technology is the incorporation of its core library in the data exploration tool TableView [10]. TableView is a web-aware 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, then make use of the embedded CLUTO library to create clusters for further exploration.

References

[1] MIT Cancer Genomics Group. Genecluster 2.0, 2002. http://www-genome.wi.mit.edu/cancer/software/genecluster2/gc2.html.

[2] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[3] Michael Eisen. Cluster 2.20 and treeview 1.60, 2002. http://rana.lbl.gov/EisenSoftware.htm.

[4] UK European Bioinformatics Institute. EPCLUST, 2003. http://ep.ebi.ac.uk/EP/EPCLUST/.

[5] S. P. Fodor, R. P. Rava, X. C. Huang, A. C. Pease, C. P. Holmes, and C. L. Adams. Multiplexed biochemical assays with biological chips. *Nature*, 364:555–556, 1993.

[6] National Center for Genome Resources. Genex, 2002. http://www.ncgr.org/genex.

[7] Genedata. Expressionist. Switzerland, http://www.genedata.com.

[8] J. Han, M. Kamber, and A. K. H. Tung. Spatial clustering methods in data mining: A survey. In H. Miller and J. Han, editors, *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis, 2001.

[9] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[10] James E. Johnson, Martina Stromvik, Kevin A. T. Silverstein, J. A. Crow, Elizabeth Shoop, and Ernest F. Retzel. Tableview: portable genomic data visualization. *Bioinformatics*, in press, 2003.

[11] G. Karypis, E.H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, 32(8):68–75, 1999.

[12] G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1), 1999. Also available on WWW at URL http://www.cs.umn.edu/~karypis. A short version appears in Intl. Conf. on Parallel Processing 1995.

[13] George Karypis. CLUTO a clustering toolkit. Technical Report 02-017, Dept. of Computer Science, University of Minnesota, 2002. Available at http://www.cs.umn.edu~cluto.

[14] Rosetta Biosoftware. Rosetta Resolver. Kirkland, WA, http://www.rosettabio.com.

[15] A. I. Saeed, V. Sharov, J. White, J. Li, W. Liang, N. Bhagabati, J. Braisted, M. Klapa, T. Currier, M. Thiagarajan, A. Sturn, M. Snuffin, A. Rezantsev, D. Popov, A. Ryltsov, E. Kostukovich, I. Borisovsky, Z. Liu, A. Vinsavich, V. Trush, and J. Quackenbush. TM4: a free, open-source system for microarray data management and analysis. *BioTechniques*, 34(2):374–378, 2003.

[16] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270, 1995.

[17] G. Sherlock, T. Hernandez-Boussard, A. Kasarskis, G. Binkley, J. Matese, S. Dwight, M. Kaloper, S. Weng, H. Jin, C. Ball, M. Eisen, P. Spellman, P. Brown, D. Botstein, and J. Chery. The stanford microarray database. *Nucleic Acids Research*, 29(1), 2001.

[18] Silicon Genetics. GeneSpring. Redwood City, CA, http://www.silicongenetics.com.

[19] SP Spanish National Cancer Center. Gene expression pattern analysis suite, 2003. http://gepas.bioinfo.cnio.es/.

[20] Spotfire. Decision Site. Somerville, MA, http://www.spotfire.com.

[21] Alexander Strehl and Joydeep Ghosh. Value-based customer grouping from large retail data-sets. In *SPIE Conference on Data Mining and Knowledge Discovery*, volume 4057, pages 33–42, 2000.

[22] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proc. of Int'l. Conf. on Information and Knowledge Management*, pages 515–524, 2002.

[23] Ying Zhao and George Karypis. Clustering in the life sciences. In M. Brownstein, A. Khodursky, and D. Conniffe, editors, *Functional Genomics: Methods and Protocols*. Humana Press, 2003.

[24] Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. *Machine Learning*, in press, 2003.