OMICS A Journal of Integrative Biology
Volume 14, Number 1, 2010

DOI: 10.1089/omi.2009.0083

ConsensusCluster: A Software Tool for Unsupervised Cluster Discovery in Numerical Data

Michael Seiler,[1] C. Chris Huang,[2] Sandor Szalma,[3] and Gyan Bhanot[1,4]

Abstract

We have created a stand-alone software tool, ConsensusCluster, for the analysis of high-dimensional single nucleotide polymorphism (SNP) and gene expression microarray data. Our software implements the consensus clustering algorithm and principal component analysis to stratify the data into a given number of robust clusters. The robustness is achieved by combining clustering results from data and sample resampling as well as by averaging over various algorithms and parameter settings to achieve accurate, stable clustering results. We have implemented several different clustering algorithms in the software, including K-Means, Partition Around Medoids, Self-Organizing Map, and Hierarchical clustering methods. After clustering the data, ConsensusCluster generates a consensus matrix heatmap to give a useful visual representation of cluster membership, and automatically generates a log of selected features that distinguish each pair of clusters. ConsensusCluster gives more robust and more reliable clusters than common software packages and, therefore, is a powerful unsupervised learning tool that finds hidden patterns in data that might shed light on its biological interpretation. This software is free and available from http://code.google.com/p/consensus-cluster.

Introduction

Clustering is a well-known machine-learning approach for pattern discovery in unknown data. It has been used in the field of Bioinformatics for diverse applications such as cancer classification and phylogenetic branch identification (Excoffier, 1990; Sorlie et al., 2001). However, many clustering algorithms have limitations, such as high computational complexity, sampling biases, sensitivity to data perturbation, etc., which limit their applicability or make it difficult to interpret and generalize the results. In particular, many methods are biased toward a particular shape in the data, and stochastic methods, such as K-Means, can have varying results depending on their initial conditions (Alpaydin, 2004). The consensus ensemble clustering algorithm attempts to mitigate these limitations of individual clustering algorithms by averaging over various clustering methods and random resamplings of the data to produce robust clusters (Monti et al., 2003; Strehl and Ghosh, 2002). Our stand-alone software package is called ConsensusCluster. It is is written in Python and C, and is supported on all major computing platforms, either in source form or as a binary (on the Win32 platform). Unlike single algorithm-based clustering software, ConsensusCluster uses the consensus ensemble clustering algorithm to combine multiple clustering methods. The program takes as input a tab-delimited text file with sample IDs located in the first row and gene IDs located in the first column. It also provides a simple Python API for creating data file parsers in the "parsers.py" module. Using this module, it is possible to create parsers for any data file format containing multiple subjects for which the feature sets are identical (e.g., all genes on a microarray chip).

As the ConsensusCluster run proceeds, clustering progress is noted in the main tab and recorded in detailed runtime log files. In addition to cluster membership information, these log files include lists of those features that best separate each pair of clusters, sorted by signal-to-noise ratio (Hengpraprohm and Chongstitvatana, 2007). When any k-clustering run is completed, each subsample dataset has been clustered into k partitions, the consensus matrix is created, the data is separated into clusters, and a clustering heatmap is placed in the working directory. If the consensus matrix is clustered using Hierarchical clustering (Hartigan, 1975), a dendrogram is also added to the output to provide additional visual information about the clustering results.

ConsensusCluster is designed to be subclassed in the Object-Oriented Programming style, and detailed instructions for subclassing are provided in the code comments. It is our

[1]BioMaPS Institute, Rutgers University, Piscataway, New Jersey.
[2]Centocor R&D Inc., Radnor, Pennsylvania.
[3]Centocor R&D Inc., San Diego, California.
[4]Department of Molecular Biology and Biochemistry & Department of Physics, Rutgers University, Piscataway, New Jersey.

intention to provide a useful framework for code expansion and enhancement, as well as to provide a stand-alone program useful to researchers. To this end, ConsensusCluster is built on standard Python scientific computing libraries, including Numpy (Ascher et al., 2001) and Matplotlib (Hunter, 2007).

Materials and Methods

The *Settings* tab (Fig. 1) provides the user with simple configuration options. Because clustering is typically run multiple times at different k values, the option "k-value range" is provided, which automatically runs the full consensus clustering process for each k in that range. Options are also provided for common normalization procedures such as mean or median centring and log2 reexpression of the data. Several clustering methods are available, which can be chosen in any combination through checkbox input or from the command line.

This software package performs each of the following actions once a data file is selected for clustering and the button labeled "Begin Clustering" is clicked.

Principal Component Analysis (PCA) feature selection

PCA (Jolliffe, 2002; Wall et al., 2003) is used to select the set of eigenvectors that represent a user-specified fraction of the total variation. Feature reduction is achieved by retaining only features whose coefficients have the largest absolute values in the selected eigenvectors. At this time a two-dimensional PCA plot projecting the data along the first two principal components is produced by ConsensusCluster (Fig. 3).

Clustering random subsets of samples and features

Often, the quality, stability, and reliability of the clusters is significantly affected by a skew in the dataset due to sampling or feature set bias. For instance, when the data represents two or more classes, and one of the classes is either oversampled or has an overabundance of features that are specific to it compared to the other classes, simple clustering analysis will be biased toward this cluster and its properties. In ConsensusCluster, for each given value of k, we perform several iterations on randomly sampled data subsets and combine results to produce a robust clustering result. At each iteration, a new dataset of user-specified size is generated by selecting a random subset of the samples, the SNPs/genes, or both. This data subset is used to determine k-clusters using the selected clustering method before a new subset is selected and clustered. The clustering results from the various data subsets are then combined to produce the final clustering results. This is akin to the machine-learning process of bootstrap aggregation, or "bagging," which is known to reduce these biases (Alpaydin, 2004).

Sample clustering

The software implements a number of clustering methods for the user to choose from, which can be selected in multiples through checkbox selection or from the command line. The methods implemented so far are as follows:

1. K-Means—clusters are partitioned around k centroids in feature space. K-Means finds local minima in sample density space and thus can have varying results depending on stochastic initial conditions. Consensus

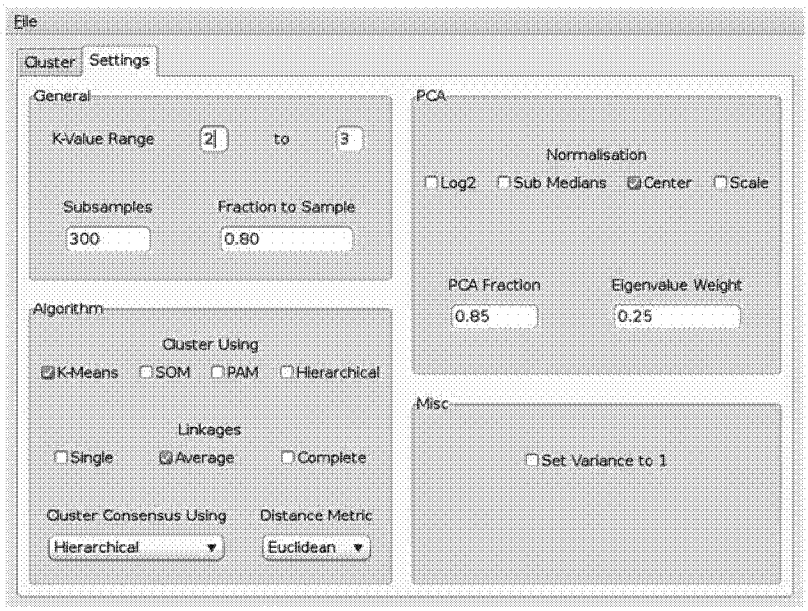

FIG. 1. *Settings* tab from the ConsensusCluster interface. "K-value range" allows the user to perform clustering operations at multiple k values automatically. "Subsamples" and "Fraction to Sample" control random dataset generation, which reduces the bias introduced by both sample and feature sets. Normalization options control basic data preprocessing procedures such as median and mean centering, whereas "PCA Fraction" and "Eigenvalue Weight" control feature selection. A number of clustering algorithms are also available for selection in any combination, including K-Means, Self-Organizing Map, Partition Around Medoids, and Hierarchical clustering methods.

clustering reduces this variance by finding the expected value of the K-Means algorithm through averaged repeats (Alpaydin, 2004).
2. Self-Organizing Map (SOM)—SOM (Kohonen, 2000) is a type of neural network that works by training nodes represented by centroids in feature space. After training, the network is used to classify the samples.
3. Partition Around Medoids (PAM)—PAM (Kaufman and Rousseuw, 1990) attempts to select k representative samples as medoids. It is robust and insensitive to outliers because it attempts to minimize the sum of dissimilarities between each sample in a cluster and its representative medoid.
4. Hierarchical Clustering—this is an agglomerative clustering method that can be run using "average," "single," or "complete" linkages (Hartigan, 1975). Clusters are determined by applying a min-cut to k clusters from the cluster tree.

Building the consensus matrix

The software combines the results from m clustering datasets for n samples into a consensus matrix M. The components $M_{(i,j)}$ are the fraction of times sample i and sample j were clustered together over all m.

Cluster the consensus matrix

The consensus matrix M is clustered using $1 - M_{(i,j)}$ as the distance metric. By default, average-linkage hierarchical clustering is used, although any algorithm that accepts a distance matrix can be used to perform the final clustering. When hierarchical clustering is selected, ConsensusCluster will generate and append a dendrogram to the consensus matrix heatmap produced in the final step.

Reorder the consensus matrix

Finally, the consensus matrix is reordered using the Simulated Annealing local optimization algorithm (Alexe et al., 2008). This procedure attempts to optimize the placement of similar samples closer to one another, and dissimilar samples farther apart. This creates an informative consensus matrix heatmap where clusters are grouped together as "boxes" along the diagonal. ConsensusCluster automatically saves this image to the current folder.

Results

ConsensusCluster provides a simple, stand-alone interface for researchers to utilize the consensus clustering algorithm on numerical data. The interface is easy to use, and requires only a tab-delimited text file containing samples and measurements, such as can be obtained by exporting, or simply copying and pasting, from an Excel spreadsheet. For exploratory research, ConsensusCluster can also be completely automated from the command line. This complex method is performed automatically without input needed from the user, enabling the user to focus on discovery.

The ConsensusCluster programme is useful for phylogenetic tree investigation independent of computationally costly methods such as maximum likelihood tree estimation. By clustering SNP data, for example, it is possible to reconstruct an unrooted tree based on cluster splits which occur at each k value used (Alexe et al., 2008). Simply using ConsensusCluster to analyze each distinguished cluster forms a hierarchical ordering of sample groups. We have analyzed 940 genome-wide autosomal SNP samples from the Human Genome Diversity Project (Li et al., 2008), and shown that ConsensusCluster can produce clusters based on polymorphisms, which, as expected, correlate well with the geographic locations where the samples were collected (Figs. 2, 3). By continuing our analysis at larger and larger k values, we can estimate the complete phylogenetic tree of these clusters (Alexe et al , 2008).

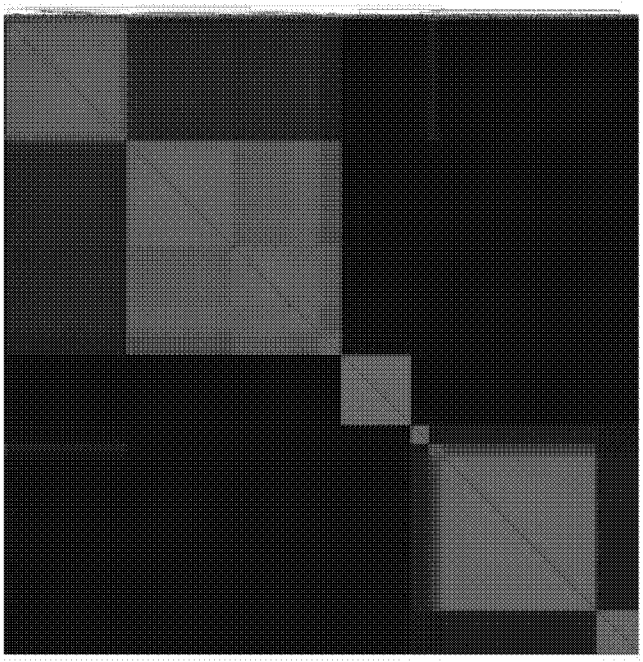

FIG. 2. Consensus matrix heatmap of 300 K-means clustering iterations, $k = 6$, produced by ConsensusCluster during operation. The consensus matrix M is produced by assigning to each $M_{i,j}$ the fraction of times sample i and sample j were clustered together. Brighter spots in the heatmap correspond to a greater likelihood of cluster association. Clusters are illustrated by "boxes" along the diagonal. The data represented are 940 genome-wide autosomal SNP samples taken from the Human Genome Diversity Project (Li et al., 2008).

This method of clustering has also been used to distinguish robust breast cancer tumor subtypes in gene microarray data (Alexe et al., 2007). Eight subtypes of breast cancer were identified, each with differing genetic signatures and pathways. These signatures were verified in the datasets of independent studies, and it was also shown that the survival characteristics and molecular signatures of the subtype were significantly distinct. Consensus clustering can be used in this way to mine clinically relevant and previously unknown disease classifications in gene expression data (Fig. 4).

Discussion

Unsupervised clustering algorithms divide data into groups such that the intracluster similarity is maximized and the intercluster similarity is minimized. Clustering methods

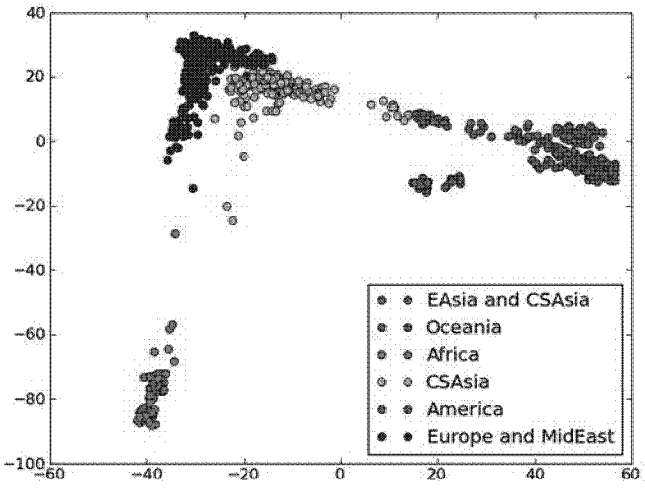

FIG. 3. PCA plot representing 940 genome-wide autosomal SNP samples taken from the Human Genome Diversity Project (Li et al., 2008). The colors indicate groups determined by ConsensusCluster using the K-means clustering algorithm over 300 iterations, $k = 6$. These groups are labeled by the geographical locations of their constituent samples.

can be divided into hierarchical, partitioning, probabilistic, and grid-based methods. The consensus clustering method (Monti et al., 2003) makes clusters by taking a weighted combination of several methods, averaging for each method over datasets obtained by bootstrapping over samples and features to improve the quality and robustness of the clusters identified, and then combining the results across all methods. The consensus cluster approach involves two steps: (1) generate a collection of clustering solutions for a given method by bootstrapping over samples and features, and (2) combine the solutions to produce a single "best" clustering of the data.

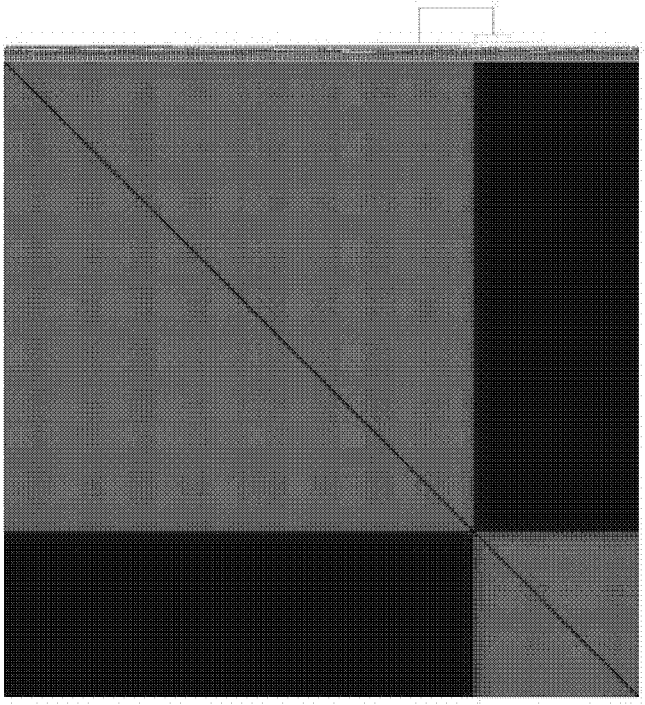

FIG. 4. ConsensusCluster clearly separates ER-positive and ER-negative breast cancer tumor microarray samples at $k = 2$. These clinical parameters define luminal-like and basal-like breast cancer tumors, respectively (Alexe et al., 2007). The data are composed from 237 HER2-samples obtained from http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE2034 (Wang et al., 2005).

There are several consensus clustering packages available at the time of this writing. One such software is available as part of the GenePattern suite of genomic analysis software (Reich et al., 2006). Both ConsensusCluster and GenePattern currently provide implementations of K-Means, Self-Organizing Map, and Hierarchical clustering methods, as well as Partition Around Medoids in the case of ConsensusCluster, and Nonnegative Matrix Factorization in the case of GenePattern. Both packages also offer resampling-based clustering to reduce data bias. However, clustering using GenePattern's implementation is limited to a single clustering algorithm, wherea ConsensusCluster allows for any combination of clustering methods to be used to reduce algorithmic bias. Also, GenePattern does not support feature selection of any kind, whereas ConsensusCluster offers PCA-based feature selection to diminish computational complexity.

Other implementations include SC2ATmd and CLUE: CLUster Ensembles (Hornik, 2005; Olex, 2007). SC2ATmd, while allowing for multiple methods to participate in clustering, currently only natively supports the K-Means and Hierarchical clustering algorithms. In addition, it does not support resampling of the input data. CLUE is an extensible framework for the R statistical software environment. As such, it does not support any clustering algorithms natively per se, but depends instead on separately available R packages for this functionality. It does, however, support a number of resampling schemes and a wide variety of cluster consensus-building algorithms. However, as a framework it cannot be used in a stand-alone manner unlike ConsensusCluster because it depends on other packages to accomplish clustering tasks.

The averaging and stability analysis implicit in the bootstrapping and consensus aggregation steps of ConsensusCluster identifies clusters that are stable to perturbations of the data, choice of clustering method, and the variance of the methods themselves. This averaging allows the identification of strong feature signatures within each cluster that would be more difficult to distinguish using simple clustering methods (Alexe et al., 2006). For breast cancer, using the ConsensusCluster approach on gene expression data, we have identified genes with a noise-independent differential expression between disease classes that allowed us to understand the biological mechanisms driving these clinically relevant subtypes of breast cancer and their correlation with patient survival, risk of recurrence, and drug efficacy (Alexe et al., 2007). These breast cancer subclasses are not distinguishable if a single clustering method is applied to the data. In addition, it is also demonstrated in these papers that different clustering methods applied to the same dataset or the same clustering method applied to subsets of the same dataset often give different clusters. This problem is resolved only by a suitable averaging over the clusters, which is accomplished in ConsensusCluster by the "consensus matrix."

In a separate application, the same method applied to single nucleotide polymorphisms (SNPs) in mitochondrial DNA sequences allowed us to develop a robust and accurate phy-

logeny of maternal ancestry that describes the migration of modern humans out of Africa. In this article, it was shown that without using consensus clustering, it was not possible to identify robust signatures for internal branch polymorphisms, resulting in trees with poor branch consensus (Alexe et al., 2008).

Conclusion

ConsensusCluster is an easy to use software package for robust clustering analysis of high-dimensional numerical data. It can be run in source form alongside any Python installation, or in stand-alone binary form on the Win32 platform. The consensus clustering algorithm used has been shown to be more accurate and more robust when compared with single clustering algorithms. ConsensusCluster is also a framework that can be easily integrated into existing Python projects. It is a powerful tool for exploratory analysis and classification of unknown data.

Acknowledgments

The work of M.S. and G.B. was partly supported by Centocor R&D, Inc. We thank Gabriela Alexe for a critical reading of the manuscript. We thank the reviewers for many excellent suggestions.

Author Disclosure Statement

No competing financial interests exist.

References

Alexe, G., Dalgin, G.S., Ramaswamy, R., Delisi, C., and Bhanot, G. (2006). Data perturbation independent diagnosis and validation of breast cancer subtypes using clustering and patterns. *Cancer Informatics* 2, 243–274.

Alexe, G., Dalgin, G.S., Scanfeld, D., Tamayo, P., Mesirov, J.P., Delisi, C., et al. (2007). High expression of lymphocyte-associated genes in node-negative Her2+ breast cancers correlates with lower recurrence rates. *Cancer Res* 67, 10669–10676.

Alexe, G., Vijaya Satya, R., Seiler, M., Platt, D., Bhanot, T., Hui, S., et al. (2008). PCA and clustering reveal alternate mtDNA phylogeny of N and M clades. *J Mol Evol* 67, 465–487.

Alpaydin, E. (2004). *Introduction to Machine Learning (Adaptive Computation and Machine Learning)* (The MIT Press, Cambridge, MA).

Ascher, D., Dubois, P.F., Hinsen, K., Hugunin, J., and Oliphant, T. (2001). *Numerical Python*. Tech. Report UCRL-MA-128569 (Lawrence Livermore National Laboratory, Livermore, CA).

Excoffier, L. (1990). Evolution of human mitochondrial DNA: evidence for departure from a pure neutral model of populations at equilibrium. *J Mol Evol* 30, 125–139.

Hartigan, J.A. (1975). *Clustering Algorithms* (John Wiley & Sons, Inc., New York).

Hengpraprohm, S., and Chongstitvatana, P. (2007). Selecting informative genes from microarray data for cancer classification with genentic programming classifier using K-means clustering and SNR ranking. In *Proceedings of the 2007 Frontiers in the Convergence of Bioscience and information Technologies* (IEEE Computer Society, Washington, DC), pp. 211–218.

Hornik, K. (2005). A CLUE for CLUster ensembles. *J Stat Software* 14, 1–25.

Hunter, J.D. (2007). Matplotlib: A 2d graphics environment. *Comput Sci Eng* 9, 90–95.

Jolliffe, I.T. (2002). *Principal Component Analysis*, 2nd ed. (Springer, Berlin).

Kaufman, L., and Rousseeuw, P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis* (Wiley-Interscience, New York).

Kohonen, T. (2000). *Self-Organizing Maps* (Springer, Berlin).

Li, J.Z., Absher, D.M., Tang, H., Southwick, A.M., Casto, A.M., Ramachandran, S., et al. (2008). Worldwide human relationships inferred from genome-wide patterns of variation. *Science* 319, 1100–1104.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Mach Learn* 52, 91–118.

Olex, A. (2007). SC2ATmd: Standard and Consensus Clustering Analysis Tool for Microarray Data. Available from: http://compbiosci.wfu.edu/tools.htm.

Reich, M., Liefeld, T., Gould, J., Lerner, J., Tamayo, P., and Mesirov, J.P. (2006). Genepattern 2.0. *Nat Genet* 38, 500–501.

Sorlie, T., Perou, C.M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., et al. (2001). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc Natl Acad Sci USA* 98, 10869–10874.

Strehl, A., and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining partitionings. In *Proceedings of AAAI 2002, Edmonton, Canada* (AAAI, Menlo Park, CA), pp. 93–98.

Wall, M.E., Rechtsteiner, A., and Rocha, L.M. (2003). *Singular Value Decomposition and Principal Component Analysis*, chapter 5 (Kluwer, Norwell, MA), pp. 91–109.

Wang, Y., Klijn, J.G., Zhang, Y., Sieuwerts, A.M., Look, M.P., Yang, F., et al. (2005). Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet* 365, 671–679.

Address correspondence to:
Gyan Bhanot
Department of Molecular Biology and Biochemistry & Department of Physics
Rutgers University
604 Allison Road
Piscataway, NJ 08854

E-mail: gyanbhanot@gmail.com