Published online 27 May 2009 *Nucleic Acids Research, 2009, Vol. 37, Web Server issue* ***W63–W67***
doi:10.1093/nar/gkp430

AutoClass@IJM: a powerful tool for Bayesian classification of heterogeneous data in biology

Fiona Achcar[1,2], Jean-Michel Camadro[2] and Denis Mestivier[1,*]

[1]'Modeling in Integrative Biology' Group and [2]'Protein Engineering and Metabolic Control' Group, Jacques Monod Institute, UMR7592 CNRS and Univ Paris-Diderot, Bâtiment Buffon, 15 rue Hélène Brion, 75205 Paris Cedex 13, France

Received January 31, 2009; Revised April 23, 2009; Accepted May 11, 2009

ABSTRACT

Recently, several theoretical and applied studies have shown that unsupervised Bayesian classification systems are of particular relevance for biological studies. However, these systems have not yet fully reached the biological community mainly because there are few freely available dedicated computer programs, and Bayesian clustering algorithms are known to be time consuming, which limits their usefulness when using personal computers. To overcome these limitations, we developed AutoClass@IJM, a computational resource with a web interface to AutoClass, a powerful unsupervised Bayesian classification system developed by the Ames Research Center at N.A.S.A. AutoClass has many powerful features with broad applications in biological sciences: (i) it determines the number of classes automatically, (ii) it allows the user to mix discrete and real valued data, (iii) it handles missing values. End users upload their data sets through our web interface; computations are then queued in our cluster server. When the clustering is completed, an URL to the results is sent back to the user by e-mail. AutoClass@IJM is freely available at: http://ytat2.ijm.univ-paris-diderot.fr/AutoclassAtIJM.html.

INTRODUCTION

High throughput experiments, such as gene expression microarrays in life sciences, result in very large data sets. In the process of analyzing such large data sets, one of the first steps most often used is to subdivide them into smaller groups of items sharing a number of common traits. Thus, clustering is frequently critical in the analysis of those data sets.

Several clustering algorithms have been proposed, including hierarchical clustering, *k*-means and S.O.M. as well as several enhancements of these algorithms (1–5). Several theoretical and applied studies have shown that unsupervised Bayesian classification systems are of particular relevance for biological studies (6–15). Therefore, these clustering algorithms are now increasingly being used in biological studies.

AutoClass is a general purpose clustering algorithm developed by the Bayesian Learning Group at the N.A.S.A. Ames Research Center since the 1980s (16,17). AutoClass is an unsupervised Bayesian classification system based upon the finite mixture model supplemented by a Bayesian method and an Expectation–Maximization algorithm for determining the optimal classes. AutoClass uses a maximum likelihood to find the class description that best predicts the data (a summary of AutoClass algorithm is presented in Supplementary Figure 1). Similar approaches have been developed using infinite mixture models and Gibbs sampling for parameters estimation (6). Our 3-year experience, based on a collaboration between a bioinformatics group and wet lab, and supported by validation of the algorithm using both simulated (see Supplementary Data) and experimental data (see below), persuaded us that AutoClass has many powerful characteristics well suited for biological data sets: (i) the user does not need to specify the number of classes, which makes this algorithm very attractive for overcoming the difficult problem of cutting hierarchical trees and selecting the correct number of clusters, (ii) AutoClass allows the user to mix heterogeneous data (discrete and real-valued) and (iii) AutoClass is able to handle missing values. Moreover, it has been designed for handling large dataset. AutoClass has been used for many years in different fields (astrophysics, economic, etc.), but by very few groups in biology (18–24).

However, as previously reported (7,10), Bayesian clustering algorithms suffer a significant decrease in computing performance as the data sets size becomes very large. End users are therefore faced with the problem of time limitation when using their personal computers. Moreover, large data sets produced by high-throughput

*To whom correspondence should be addressed. Tel: +33 (0)157278055; Fax: +33 (0)157278101; Email: mestivier@ijm.jussieu.fr

data acquisition dramatically increase this 'need for computation time'.

To our knowledge, very few web servers implement Bayesian clustering algorithms (11,25) and none of them implement as many characteristics as AutoClass.

To overcome these limitations, we make available, free of charge to the academic community, computational resources and we developed AutoClass@IJM, a web interface dedicated to the AutoClass algorithm.

IMPLEMENTATION

We implemented AutoClass-c, version 3.3.4 on a Unix system. The cluster server is composed of 18 blades (64bits bi-processors (uni- or dual-core), 2Go RAM) summing up to a total of 64 CPU units, which offers end users a high computer power. OpenPBS (Torque/Maui) is used to schedule the queued jobs.

WEB APPLICATION DESCRIPTION

Using AutoClass@IJM requires two steps: (i) preparing the data and (ii) submitting data files (with optional modifications of default clustering parameters). An URL to the results is sent back to the user by e-mail. The return time can vary from minutes/hours to days depending on the size of data set and the cluster load.

PREPARING DATA FILES

AutoClass can handle three different types of data: (i) singly bounded real numbers (Real Scalar as named by AutoClass), such as length, weight, etc., (ii) real numbers distributed on the two sides of an origin (Real location), such as Cartesian coordinates (in this case, the origin is 0.0), microarray log ratio, elevation (where sea level is the origin), etc. and (iii) discrete data: any qualitative data, such as chromosome number, phenotype, eyes color, etc. For each type of data, the web interface provides a specific input field.

- The user must prepare a different text file for each type of data he wants to classify.
- Each file must be tab-delimited and contain at least two columns: the first one with the ID (shorter than 30 characters) of the elements to cluster (such as gene names, etc.) and the other columns with the data (columns with a unique value for all lines are prohibited and must be removed). Each file may contain a header line. AutoClass can handle an 'unlimited' number of lines and a maximum of 999 columns (default setting of AutoClass; the maximum number of columns may be increased upon user's request).
- In each file, elements with the same ID will be detected and automatically processed as a single entry with: (i) the mean value for numerical data, (ii) the last value for discrete data.
- The same ID in different files represents the same element.
- Missing values are allowed (blank fields in the data file).

We provide two examples on the web interface (data sets and result files) as illustrative material: one data set with real valued data, and one with heterogeneous data (discrete and real valued data).

SUBMITTING DATA FILES

The user must provide an e-mail address and upload his data files in the appropriate fields. AutoClass uses several parameters: we provide an optimized default set. The default parameters are AutoClass defaults except for the 'max_n_cycles' parameters (the maximum number of cycles). AutoClass choose the best among 100 classifications. Each classification is performed as a recursive process: a classification stops if the convergence criteria are met or if the maximum number of cycles is reached (see Supplementary Figure 1). Gene expression data are especially difficult to cluster for they are very noisy. Therefore, AutoClass default maximum number of cycle (200) is reached too often according to our experience. We thus decided to set this parameter to 1000 in order for most classifications to converge before the maximum number of cycle is reached.

However, the user may change the 'error' parameter. This error is relative (i.e. the ratio of the error to the value) for real scalars, and it is a constant for real location values.

Each analysis is submitted as a single job. Once submitted, the jobs are queued, and when the job starts running, a first e-mail is send to the user.

ERROR PARAMETER

The error parameter applies to the inputted data. As quoted in the AutoClass documentation files:

> *The fundamental question in all of this is: "**to what extent do you believe the numbers that are to be given to AutoClass?**" AutoClass will run quite happily with whatever it is given. It is up to the user to decide what is meaningful and what is not.*

For practical purposes: (i) when dealing with 'real scalars' data type, the error parameter is expressed as percent of the value and should obviously be less than 1 (i.e. <100%). (ii) when dealing with 'real location' data type, the error parameter is a constant value.

If the error parameter entered by user is too large with respect to the data, the error message generated by AutoClass is interpreted and an e-mail is sent to the user with AutoClass log file attached.

OUTPUT DATA

AutoClass computes classes using all the inputted data. After completion of the job, a single zip archive is generated containing: (i) a tab-delimited file that associates each ID with the index of its class, (ii) two CDT files to read the results in JavaTreeview-like software (26) (*if input data are exclusively numerical, otherwise use your favorite spreadsheet to read the file)*: one contains the experimental data and the probabilities for each item to belong to

different classes; the second contains only the experimental data (to help visual identification of classes, blank lines are introduced between classes in the CDT files) (iii) a log file from AutoClass. A second e-mail which contains an URL to the zip archive is then sent for the user to upload his results.

Early development of AutoClass@IJM used our own yeast microarrays (and yeast data from public databases). Therefore, when the IDs are yeast standard ORF names according to the *Saccharomyces* Genome Database (SGD) nomenclature (27), the CDT output files will be annotated with SGD ORF descriptions. Otherwise, the CDT output files contain only the initial IDs.

SOME BENEFITS OF USING OF BAYESIAN CLASSIFICATION: AN EXAMPLE ON MICROARRAY DATA

We provide a typical example of classification in Figure 1. The entire clustering of the dataset from Yoshimoto *et al.* (28) using AutoClass shows that homogeneous classes were computed. The original paper emphasized on the characterization of yeast genes responsive to calcium in a calcineurin-dependent pathway and thus to better characterize the Crz1p-binding sites. Using AutoClass, we were able to identify classes of Crz1 responsive genes, from which the Crz1p DNA-binding consensus sequence [AC][AC]GCC[AT]C was extracted using MEME (29) (*E*-values from 10^{-8} to 10^{-13}). We also characterized other classes of genes of interest when dealing with calcium and sodium response in yeast such as a class of genes strongly repressed by both calcium and sodium. Analysis of the promoter region of these genes in this class led us to find the PAC and RRPE-binding sites described by Beer and Tavazoie (30) (*E*-value = 4.3×10^{-225}). Interestingly, these genes do not belong to the 'ribosome biogenesis regulon' described by these authors. We are currently investigating the mechanisms of regulation of this group of genes. This group of gene was not discussed in Yoshimotos's paper (28), and is a good example of the power of Bayesian clustering as a discovery tool.

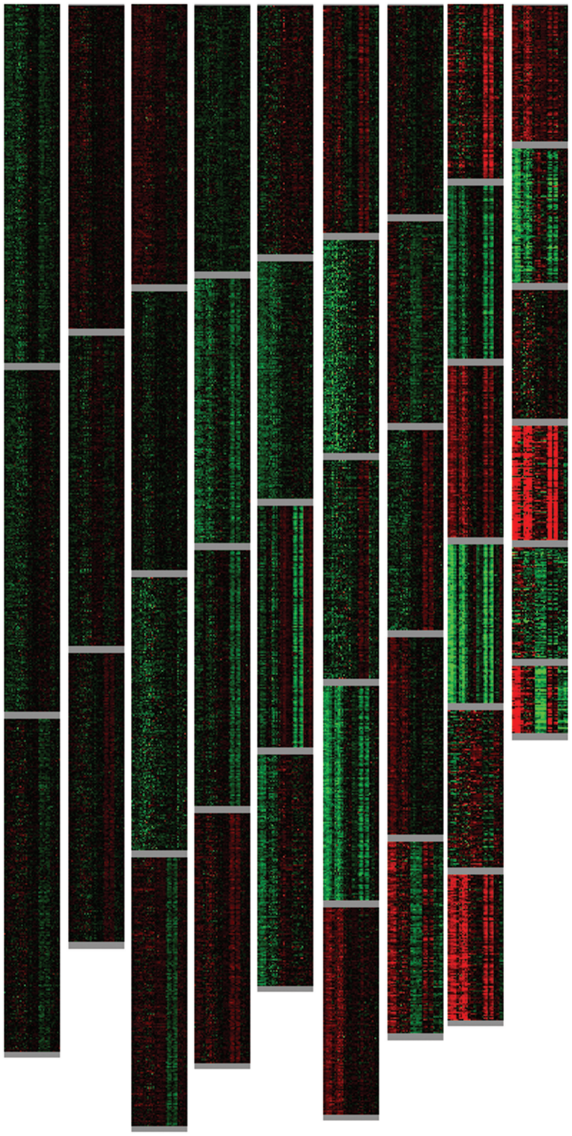

Figure 1. Overview of JavaTreeView output from AutoClass clustering of representative yeast microarray data (6300 rows, 35 columns; Gene Expression Omnibus database at NCBI: GSE3456). To help visual identification of classes, blank lines are introduced between classes.

EXECUTION TIME

Since Bayesian classification is demanding in computation time, it is worthwhile for an end-user to know the time required for a given analysis. Such estimation is difficult due to the numerous parameters involved: (i) computer specifications, such as hardware, task scheduling, load balancing of the cluster and (ii) the underlying structure of the data. In order to provide a estimation, we used several published data sets to generate two large compendiums of data: 6000 rows and 120 columns. Each data set was segmented into subsets of increasing size (from 500 rows to 6000 and from 5 columns to 120). Each subset was classified three times and we report in Figure 2 the mean computation time (in seconds) required by AutoClass on our system.

Typically, the data set of 500 rows and one column was classified in 16 s, the data set of 6000 rows and 10 columns was classified in 1 h and the entire data set (6000 rows and 120 columns) was classified in about 16 h. This illustrates the difficulty an end-user might encounter when using his own personal computer for clustering, and is for us a strong motivation to make appropriate computational resources with an easy-to-use web interface available to the community.

DISCUSSION AND CONCLUSION

High throughput experiments generating large data sets have recently pinpointed the need for biologists to have access to computational resources dedicated to the clustering of their data. These data sets may be heterogeneous in nature (discrete and real values) and often have

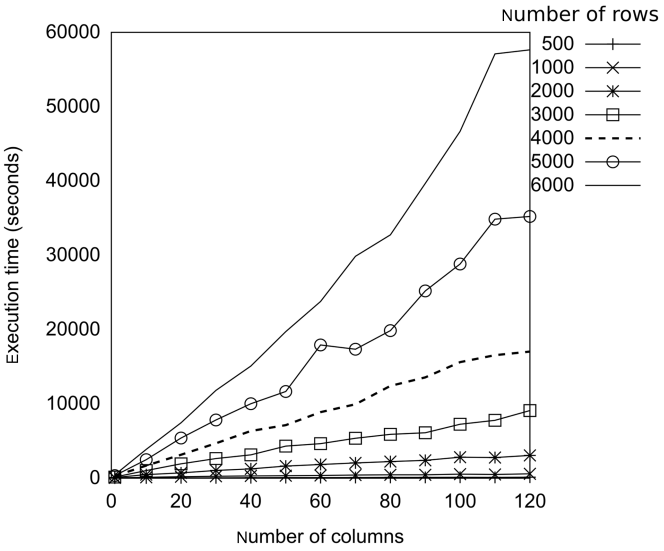

Figure 2. Execution time (in seconds) of AutoClass for various data set sizes. Horizontal axis: Number of columns of the data set; vertical axis: mean execution time (in seconds—see text). Each curve represents a different number of rows in the data set.

missing values. Since Bayesian clustering algorithms are now recognized as highly valuable alternative to standard clustering algorithms, we anticipate that making such algorithm available for use may provide much added-value through the possibility to classify and analyze complex aggregated data sets in a single process, a situation often found when dealing with biological data. We show in this article that the general drawback of Bayesian algorithms in terms of computational cost may be as high as tens of hours for 'medium size' data sets (6000×120 matrices) and this computation time may rise when analyzing data from e.g. human transcriptome where the expression patterns of more than 35 000 rows is clustered. The figures show that the required computation time may rapidly be out of range when using a personal computer, a standard situation in many biology labs.

We developed and offer the academic community a server dedicated to the well-validated Bayesian clustering algorithm, AutoClass. Our web server provides a high computer power to end-users and is an integrated tool to aggregate and classify in a single process heterogeneous data. Data files are simple tab-delimited text files that can easily be generated by the user from raw data.

As data sets size continuously increases, future developments will focus on implementing (i) tools that allow the user to supervise his running jobs and (ii) the parallel version of AutoClass (31).

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

French MESR grant-in-aid (to F.A.). Funding for open access charge: CNRS UMR7592 – Université Paris Diderot.

Conflict of interest statement. None declared.

REFERENCES

1. Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.
2. Gollub,J. and Sherlock,G. (2006) Clustering microarray data. *Methods Enzymol.*, **411**, 194–213.
3. De Hoon,M.J., Imoto,S., Nolan,J. and Miyano,S. (2004) Open source clustering software. *Bioinformatics*, **20**, 1453–1454.
4. Dembele,D. and Kastner,P. (2003) Fuzzy c-means method for clustering microarray data. *Bioinformatics*, **19**, 973–980.
5. Kraj,P., Sharma,A., Garge,N., Podolsky,R. and Mcindoe,R.A. (2008) Parakmeans: Implementation of a parallelized k-means algorithm suitable for general laboratory use. *BMC Bioinformatics*, **9**, 200.
6. Medvedovic,M. and Sivaganesan,S. (2002) Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics*, **18**, 1194–1206.
7. Medvedovic,M., Yeung,K.Y. and Bumgarner,R.E. (2004) Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, **20**, 1222–1232.
8. Yeung,K.Y., Fraley,C., Murua,A., Raftery,A.E. and Ruzzo,W.L. (2001) Model-based clustering and data transformations for gene expression data. *Bioinformatics*, **17**, 977–987.
9. Ng,S.K., Mclachlan,G.J., Wang,K., Ben-tovim Jones,L. and Ng,S.W. (2006) A mixture model with random-effects components for clustering correlated gene-expression profiles. *Bioinformatics*, **22**, 1745–1752.
10. Qin,Z.S. (2006) Clustering microarray gene expression data using weighted chinese restaurant process. *Bioinformatics*, **22**, 1988–1997.
11. Xiang,Z., Qin,Z.S. and He,Y. (2007) Crcview: a web server for analyzing and visualizing microarray gene expression data using model-based clustering. *Bioinformatics*, **23**, 1843–1845.
12. Fraley,C. and Raftery,A.E. (1998) How many clusters? Which clustering method? Answers via model-based cluster analysis. *Comput. J.*, **41**, 578–588.
13. Haughton,D., Legrand,P. and Woolford,S. (2009) Review of three latent class cluster analysis packages: latent gold, polca, and mclust. *Am. Statistician*, **63**, 81–91.
14. Tadesse,M.G., Sha,N. and Vannucci,M. (2005) Bayesian variable selection in clustering high-dimensional data. *J. Am. Stat. Assoc.*, **100**, 602–617.
15. Baladandayuthapani,V., Mallick,B.K., Hong,M.Y., Lupton,J.R., Turner,N.D. and Carroll,R.J. (2008) Bayesian hierarchical spatially correlated functional data analysis with application to colon carcinogenesis. *Biometrics*, **64**, 64–73.
16. Cheeseman,P., Kelly,J., Self,M., Stutz,J., Taylor,W. and Freeman,D. (1988). AutoClass: A Bayesian Classification System, in *Proceedings of the Fifth International Conference on Machine Learning*. Morgan Kaufmann Publishers, San Francisco.
17. Cheeseman,P. and Stutz,J. (1996) Bayesian Classification (AutoClass): Theory and Results. In Fayyad,U., Piatelsky-Shapiro,G., Smyth,P. and Uthurusamy,R. (eds), *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press, Cambridge.
18. Moler,E.J., Radisky,D.C. and Mian,I.S. (2000) Integrating naive Bayes models and external knowledge to examine copper and iron homeostasis in S. Cerevisiae. *Physiol. Genomics*, **4**, 127–135.
19. Chow,M.L., Moler,E.J. and Mian,I.S. (2001) Identifying marker genes in transcription profiling data using a mixture of feature relevance experts. *Physiol. Genomics*, **5**, 99–111.
20. Semeiks,J.R., Grate,L.R. and Mian,I.S. (2005) Text-based analysis of genes, proteins, aging, and cancer. *Mech. Ageing Dev.*, **126**, 193–208.
21. Semeiks,J.R., Rizki,A., Bissell,M.J. and Mian,I.S. (2006) Ensemble attribute profile clustering: discovering and characterizing groups of genes with similar patterns of biological features. *BMC Bioinformatics*, **7**, 147.
22. Hunter,L. and States,D. (1992) Bayesian classification of protein structure. *IEEE Intell. Sys.*, **7**, 67–75.

23. Okada,Y., Sahara,T., Mitsubayashi,H., Ohgiya,S. and Nagashima,T. (2005) Knowledge-assisted recognition of cluster boundaries in gene expression data. *Artif. Intell. Med.*, **35**, 171–183.
24. Crook,A.C., Baddeley,R. and Osorio,D. (2002) Identifying the structure in cuttlefish visual signals. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, **357**, 1617–1624.
25. Rasmussen,M.D., Deshpande,M.S., Karypis,G., Johnson,J., Crow,J.A. and Retzel,E.F. (2003) Wcluto: a web-enabled clustering toolkit. *Plant Physiol.*, **133**, 510–516.
26. Saldanha,A.J. (2004) Java Treeview – extensible visualization of microarray data. *Bioinformatics*, **20**, 3246–3248.
27. Weng,S., Dong,Q., Balakrishnan,R., Christie,K., Costanzo,M., Dolinski,K., Dwight,S.S., Engel,S., Fisk,D.G., Hong,E. *et al.* (2003) Saccharomyces Genome Database (SGD) provides biochemical and structural information for budding yeast proteins. *Nucleic Acids Res.*, **31**, 216–218.
28. Yoshimoto,H., Saltsman,K., Gasch,A.P., Li,H.X., Ogawa,N., Botstein,D., Brown,P.O. and Cyert,M.S. (2002) Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in Saccharomyces Cerevisiae. *J. Biol. Chem.*, **277**, 31079–31088.
29. Bailey,T.L., Williams,N., Misleh,C. and Li,W.W. (2006) MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res.*, **34**, W369–W373.
30. Beer,M.A. and Tavazoie,S. (2004) Predicting gene expression from sequence. *Cell*, **117**, 185–198.
31. Pizzuti,C. and Talia,D. (2003) P-autoclass: scalable parallel clustering for mining large data sets. *IEEE Trans. Knowledg Data Eng.*, **15**, 629–641.