BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 9 2006, pages 1154–1156
doi:10.1093/bioinformatics/btl074

Data and text mining

GALGO: an R package for multivariate variable selection using genetic algorithms

Victor Trevino and Francesco Falciani*

School of Biosciences, University of Birmingham, Birmingham, B15 2TT, UK

Received on December 9, 2005; revised on February 13, 2006; accepted on February 24, 2006
Advance Access publication March 1, 2006
Associate Editor: Alfonso Valencia

ABSTRACT

Summary: The development of statistical models linking the molecular state of a cell to its physiology is one of the most important tasks in the analysis of Functional Genomics data. Because of the large number of variables measured a comprehensive evaluation of variable subsets cannot be performed with available computational resources. It follows that an efficient variable selection strategy is required. However, although software packages for performing univariate variable selection are available, a comprehensive software environment to develop and evaluate multivariate statistical models using a multivariate variable selection strategy is still needed. In order to address this issue, we developed GALGO, an R package based on a genetic algorithm variable selection strategy, primarily designed to develop statistical models from large-scale datasets.

Availability: GALGO can be downloaded from http://www.bip.bham.ac.uk/bioinf/galgo.html

Contact: vtrevino@itesm.mx; f.falciani@bham.ac.uk

Supplementary information: Supplementary data are available at http://www.bip.bham.ac.uk/bioinf/galgo.html

1 INTRODUCTION

Functional Genomics technologies allow us to measure thousands of transcripts, proteins and metabolites in single experiments. In order to make sense of these complex datasets, methods to select relatively small subsets of biological measurements that are associated to cell function are essential. In this context, many statistical modeling techniques have been applied to microarray data (Dudoit and Fridlyand, 2003). These approaches can be subdivided into univariate and multivariate. Univariate approaches test one feature at a time for their ability to discriminate a dependent variable. The top most significant features are then used to develop a statistical model (for an extensive comparison of classification methods in the context of a univariate variable selection strategy see Lee *et al*., 2005). Multivariate approaches takes into consideration that variables are influencing a biological outcome in the context of networks of interacting genes rather than in isolation and can take into considerations synergy between genes, proteins or metabolites. Although these approaches have been very successful there are still issues in the development of multivariate models from large datasets. These issues are related to the extremely large number of possible models that would need to be evaluated to identify the most predictive. In order to address these issues, stochastic search strategies have been developed and tested on functional genomics datasets. Among these, Markov chain Monte Carlo and Genetic Algorithms (GAs) procedures have been used successfully in the analysis of microarray data (Li *et al*., 2001; Ooi and Tan, 2003; Sha *et al*., 2004). Although a comparative study of these methods, analyzing microarray data, has not been published, the GA procedure seems to be computationally more efficient. The GA procedure starts from a random populations of models and evolves good local solutions by mimicking the process of natural selection using mechanisms such as higher rate of replication of the more effective variable subsets (chromosomes), mutation to generate variants and crossover to improve combinations (Goldberg, 1989). The fact that sets of variables (arranged in chromosomes) are tested in combination during the selection process ensures a truly multivariate variable selection. Present implementation of GA in functional genomics (Li *et al*., 2001; Ooi and Tan, 2003) is limited to a specific GA procedure, classification methods and error estimation strategy. Moreover, their output is a plain text file that is often difficult to interpret. In order to address these issues, we have developed GALGO. The package has been implemented in an object-oriented fashion, coupled with a general fitness function to guide the variable selection. In this initial release, we included fitness functions for solving classification problems and provide the documentation for implementing *ad hoc* fitness functions to solve general optimization problems. In the Supplementary Material we also compare the accuracy of models selected with GALGO with a number of methods that uses a univariate variable selection strategy. Our results support the use of GALGO as a useful tool in the analysis of large scale functional genomics data.

2 IMPLEMENTATION

The GALGO package has been conceived as an implementation of GA in object-oriented paradigm under the R language. GALGO uses a GA procedure for selecting models with a high fitness value and implements functions for the analysis of the populations of selected models as well as functions to reconstruct and characterize representative summary models (Li *et al*., 2001).

The GA procedure and GALGO object-oriented design

GALGO uses GA for selecting variable subsets. The procedure starts from a random population of variable subsets of a given size (defined as chromosomes). Each chromosome is assessed for its ability to predict a dependent variable and has a certain level of

*To whom correspondence should be addressed.

accuracy. The general principle is to replace the initial population with a new population that includes variants of chromosomes with higher classification accuracy and to repeat the process enough times to achieve a desired level of accuracy. The progressive improvement of chromosome population is driven by a number of operators that mimic the process of natural selection (selection, mutation and crossover). In order to increase the proportion of the solution space that is explored, independent chromosomes populations can be evolved in partially isolated environments (niches). Chromosomes can occasionally migrate from one niche to another ensuring that particularly good solutions can recombine. The collection of niches is called world. A detailed description of the procedure is available in the Supplementary Material. The object design of the GALGO package reflects the structure we described above. In GALGO, Gene object represents a variable whereas the Chromosome object stands for a set of n variables that will be included in the multivariate model, which will be evaluated using a fitness function. A Niche object organizes chromosomes in populations whereas the World object includes several niches. The Galgo object arranges these objects, implements the GA evolutionary process and saves the best chromosome. Finally, a BigBang object collects the result of several searches for further analysis. These objects have properties that allow users to control the process. We included most common GA operators as Reproduction, Mutation, Crossover, Migration and Elitism as methods. An important characteristic of GALGO is that the user can add custom defined properties to add new functionality.

Classification methods

In the initial release, we have included parametric and non-parametric classification methods such as *k*-nearest-neighbors, discriminant functions, nearest centroid, support vector machines, classification trees and neural networks. The first three were implemented in C whereas the others were adapted from original R packages. In the package manual we also include the code to develop models using Random Forest. RF is a decision tree-based method that uses a boosting and bagging error estimation procedure that has been demonstrated to be very effective to avoid overtraining (Dudoit and Fridlyand, 2003).

3 APPLICATION

In this paragraph we describe a typical application of GALGO. The analysis protocol has been subdivided into four steps.

Step 1: setting-up the analysis

In this initial stage of the analysis the user specifies the input data, the dependent variable (e.g. class labels), the statistical model, the desired accuracy (fitness), the error estimation scheme and the parameters that define the GA search environment. Gene expression values can be provided in a common text file or as a matrix object, which may be the result of pre-processing using other R tools such as Bioconductor (Gentleman *et al*., 2004). The classification method may be one of the six already implemented supervised classification methods, or a user-defined function. The error estimation can be defined at two levels: the classic training and test validation strategy using a single or multiple random splits, and inside the training process using *k*-fold cross-validation, random splits or re-substitution error (reviewed in Dudoit and Fridlyand, 2003). The common GA parameters are automatically configured but can also be specified.

Step 2: searching for relevant multivariate models

Every evolutionary cycle in the GA procedure starts from a random population of chromosomes and may lead to a diverse collection of good local solutions. For this reason a sufficiently large number of chromosomes should be selected in order to have a good representation of the solution space. Ideally such number should be sufficiently large to ensure that all solutions that can be found with the GA procedure are represented in the population of selected chromosomes. In order to make this possible we have designed two real time monitors that provide information on the chromosome composition, the level of convergence of the solutions and the evolution of the fitness values. These diagnostic plots are useful tools to assess when the searches converge to a stable population that can then be analyzed further.

Step 3: refinement and analysis of the population of selected chromosomes

The chromosomes selected from the GA procedure have a fixed length defined at the first step of the analysis. Although the models have the desired classification accuracy, there is a possibility that not all genes included in the model contribute significantly to the fitness value. We have implemented a backward selection strategy to derive a chromosome population where only genes that effectively contribute to the classification accuracy of the model are included (refinement). This function can also be used within the selection process in Step 2.

GALGO implements a number of functions for the analysis of the chromosome populations. These produce a text or graphical output and describe (1) the occurrence of genes in the model population, (2) the model gene composition, (3) the model accuracy, (4) the relative importance of genes in the models, (5) the evolution of the fitness function during chromosome selection and (6) prediction of new samples. In addition, gene signatures associated to specific chromosomes can be visualized using two dimensional clustering heat maps, principal component analysis plots, gene profiles or class profiles.

Step 4: Development of a representative statistical model

The aim of this part of the analysis is to develop a single representative model from the population of selected chromosomes. In order to do so, we have implemented a forward selection strategy based on the step-wise inclusion of the most frequent genes represented in the chromosome population (Li *et al*., 2001). Every model developed with GALGO can be stored and used to predict the identity of novel samples.

4 DISCUSSION

GALGO is a user-friendly R package designed for developing multivariate statistical models using large-scale 'omics' data. In the context of multivariate variable selection in large-scale datasets GALGO performs well and does not require any coding. For a more general use, its object-oriented structure allows the definition of new methods by simply recoding the fitness function. GALGO allows the development and analysis of statistical models using a unique wrapping function. These characteristics make GALGO an

ideal environment for both Bioinformaticians and computer minded biologists. The availability of a very broad spectrum of R libraries with general statistical (CRAN) or with specific machine learning functionality (such as MLinterfaces and ipred) makes GALGO an ideal prototyping environment for any analysis method that utilizes GAs as a search strategy. In conclusion, GALGO is a valuable, robust and easy to use tool for developing multivariate statistical models using multivariate variable selection.

ACKNOWLEDGEMENTS

The authors would like to thank Russell Compton, Fernando Ortega and Dov Stekel for their helpful comments. V.T. is a recipient of a fellowship from the Darwin Trust of Edinburgh.

Conflict of Interest: none declare.

REFERENCES

Dudoit,S. and Fridlyand,J. (2003) Classification in microarray experiments. In Speed,T.P. (ed.), *Statistical Analysis of Gene Expression Microarray Data*, Chapman & Hall/CRC, Boca Raton, FL, pp. 93–158.

Gentleman,R.C. *et al*. (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*., **5**, R80.

Goldberg,D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, Vol. xiii, p. 412.

Lee,J.W. *et al*. (2005) An extensive comparison of recent classification tools applied to microarray data. *Comput. Stat. Data Anal*., **48**, 869–885.

Li,L.P. *et al*. (2001) Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, **17**, 1131–1142.

Ooi,C.H. and Tan,P. (2003) Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Bioinformatics*, **19**, 37–44.

Sha,N. *et al*. (2004) Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage. *Biometrics*, **60**, 812–819.