BIOINFORMATICS

Vol. 00 no. 00 2014
Pages 1–2

switchBox: An R package for k-Top Scoring Pairs (*k*TSP) classifier development

Bahman Afsari [1,*], Elana J. Fertig [1], Donald Geman [2] and Luigi Marchionni [1,*]

[1] Department of Oncology, Sidney Kimmel Comprehensive Cancer Center, School of Medicine, Johns Hopkins University, Baltimore, MD 21205
[2] Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Summary k-Top scoring pairs (kTSP) is the aggregation of voting among two-feature switch decision rules. Each switch decides based on the reversal of the ordering of the value of two features (e.g. expression of two genes between samples from the two groups being predicted.). kTSP, like its predecessor TSP, is a parameter free classifier relying only on the relative ordering of the values of a small subset of features. Hence, kTSP classifiers are robustness to noise and potentially interpretable in biological systems. In contrast to TSP, kTSP has comparably high accuracy to standard genomics classification techniques, including Support Vector Machines (SVM) and Prediction Analysis for Microarrays (PAM).. Here, we describe switchBox package which provides R functions for finding kTSP with minimum parameters.

Availability: The switch box package is freely available from Bioconductor: Bioconductor: http://www.bioconductor.org

Contact: bahman@jhu.edu

1 INTRODUCTION

Finding gene expression biomarkers for diagnosis or prognosis has been extensively studied in numerous diseases. However, the mature clinical application of these biomarkers is scarce, particularly for human cancers. In (1), technological, mathematical and translational barriers are held responsible for this lack of clinical trials. Basing the prediction solely on the ordering of a small number of features (e.g. gene expressions), known as ranked based methodology, may overcome such barriers to clinical translation (2).

Ranked-based methods have been shown to be robust to data normalization and rise to more transparent decision rules. The first and simplest of such methodologies, the Top-Scoring Pair (*TSP*) classifier, was introduced in (3) and is based on reversal of two features (e.g. the expressions of two genes). Multiple extensions were proposed afterwards, e.g. (4; 5) and many of these extensions have been successfully applied for diagnosis and prognosis of cancer such as differentiation between two types of gastrointestinal cancer (6), predicting treatment response in breast cancer (7), and recently simplifying clinical biomarkers (8). A popular successor of *TSP* classifiers is *kTSP* (5) which applies the majority voting among multiple of the the reversal of pairs of features. In addition to being applied by peer scientists, *kTSP* shown its power by wining the ICMLA the challenge for cancer classification in the presence of other competitive methods such as Support Vector Machines (SVM) (9).

Here, we introduce an R package for *kTSP*, “switchBox.” This package chooses the gene pairs for the “kTSP” decision rule. The package also chooses the number of pairs in a novel way introduces in (10). The new method, based on the analysis of variance, is less computational intensive, mathematically more elegant, and less prone to over-fitting compared to the original method introduced in (5) and implemented in R as (11) which used an inner loop of cross-validation for tuning parameters. The package also provides more flexible feature and candidate pairs selection.Also, “switchBox” has a method for calculating the pair-wise score which may be useful outside of the classification problem like in (12). For definition of the score see Methods section.

2 METHODS

kTSP decision is based on k feature (e.g. gene) pairs, say, $\Theta = \{(i_1, j_1), \ldots, (i_k, j_k)\}$. If we denote the feature profile with $\underline{X} = (X_1, X_2, \ldots)$, the family of rank based classifiers is an aggregation of the comparisons $X_{i_l} < X_{j_l}$. Specifically, the kTSP statistics can be written as:

$$\kappa = \{\sum_{l=1}^{k} I(X_{i_l} < X_{j_l})\} - \frac{k}{2}, \tag{1}$$

where I is the indicator function. The kTSP classification decision can be produced by thresholding the κ, i.e. $\hat{Y} = I\{\kappa > \tau\}$ provided the labels $Y \in \{0, 1\}$. The standard threshold is $\tau = 0$, equivalent to majority voting. The only parameters required for calculating κ is the number and choice of feature pairs. In the introductory paper to kTSP (5), the authors proposed an ad-hoc method for feature selection. This method was based on score for each pair of features which measures how discriminative is a comparison of the feature values. If we denote the score related to the gene i and j by s_{ij}, then the score was defined as

$$s_{ij} = |\mathrm{P}\left(X_i < X_j | Y = 1\right) - \mathrm{P}\left(X_i < X_j | Y = 0\right)|.$$

[*] to whom correspondence should be addressed

© Oxford University Press 2014.

We can sort the pairs of genes by this score. A pair with large score (close to one) indicates that the reversal of the feature value predicts the phenotype accurately.

In (10), an analysis of variance was proposed for gene selection in kTSP and other rank-based classifiers. This method finds the feature pairs which make the distribution of κ under two classes *far apart* in the analysis of variance sense. In mathematical words, we seek the set of feature pairs, Θ^*, that

$$\Theta^* = \arg\max_{\Theta} \frac{\mathrm{E}\left(\kappa(\Theta)|Y=1\right) - \mathrm{E}\left(\kappa(\Theta)|Y=0\right)}{\sqrt{\mathrm{Var}\left(\kappa(\Theta)|Y=1\right) + \mathrm{Var}\left(\kappa(\Theta)|Y=0\right)}}. \quad (2)$$

This method automatically chooses the number of features and hence, it is almost a parameter free method. However, the search for Θ is very intensive search. So, a greedy and approximate search was proposed to find the optimal set of gene pairs. Interesting, the greedy search requires the calculation of the score and shows why the original kTSP (5) performed well.

3 IMPLEMENTATION

The “swithBox” package calculates the score for all eligible pairs and then sort them and finds the top disjoint pairs. At the end, it finds the number of pairs using the analysis of variance. Hence, the basic methods of the package is calculating the score for the pairs. The methods for calculating scores were developed in C to speed up the calculations. The memory allocation grows linearly with the number of pairs, which may require feature filtering in high dimensional data. If the user wishes to directly calculate the score of a desired set of features or feature pairs, they can invoke `SWAP.CalculateSignedScore` function.

The package provides a training function (`SWAP.KTSP.Train`) for the classifier and classification function (`SWAP.KTSP.Classify`) for predicting the label of an unseen sample. The training function has the flexibility for a user-defined filtering the features or the pair of features. Using this property the user can reduce the variance of the decision rules. The package also offers a function to calculate the pairwise scores of a subset of features or a subset of feature pairs estimated from training data.

(`SWAP.KTSP.Statistics`) is another useful function which provides *kTSP* statistics, that is κ in eq. 1, from a trained classifier. This method is useful for generating ROC curve, a measure of the reliance of the prediction. We have made the method flexible so that other rank-based combinations can be calculated like Fig 1 where we calculated all comparison votes.

Here is a sample code which uses all of these functions:

```
> data(matTraining);trainingGroup <– factor(gsub(”.+\\.”,””, colnames(matTraining)))
#load training data
> scores <– SWAP.CalculateSignedScore(matTraining, trainingGroup)
#Calculate scores of all pairs of genes in training set.
> classifier <– SWAP.KTSP.Train(matTraining, trainingGroup, krange=c(3:15))
#Train kTSP. ”krange” contains the candidate range for number of pairs
> trainingPrediction <– SWAP.KTSP.Classify(matTraining, classifier)
# kTSP prediction on the training set.
> kappa <– SWAP.KTSP.Statistics( matTraining, classifier)# calculating kappa
```

Acknowledgement
Funding: LM was supported by the National Institutes of Health (NIH-)NCI grant P30 CA006973; LM and EJF by the Cleveland Foundation The Helen Masenhimer Fellowship Award; BA and DG by the NIH-NCRR Grant UL1 RR 025005. EJF and BA by NIH-NCI grant K25 CA141053.

REFERENCES

[1]R. Winslow, N. Trayanova, D. Geman, and M. Miller, “The emerging discipline of computational medicine,” *Science Translational Medicine*, vol. 4, no. 158, p. 158rv11, 2012.

[2]J. A. Eddy, J. Sung, D. Geman, and N. D. Price, “Relative expression analysis for molecular cancer diagnosis and prognosis,” *Technology in cancer research & treatment*, vol. 9, no. 2, p. 149, 2010.

[3]D. Geman, C. d’Avignon, D. Naiman, *et al.*, “Gene expression comparisons for class prediction in cancer studies,” in *Proceedings 36’th Symposium on the Interface: Computing Science and Statistics*, 2004.

[4]X. Lin, B. Afsari, L. Marchionni, *et al.*, “The ordering of expression among a few genes can provide simple cancer biomarkers and signal brca1 mutations,” *BMC Bioinformatics*, vol. 10, no. 256, 2009.

[5]A. C. Tan, D. Q. Naiman, L. Xu, *et al.*, “Simple decision rules for classifying human cancers from gene expression profiles,” *Bioinformatics*, vol. 21, no. 20, pp. 3896–3904, 2005.

[6]N. Price, J. Trent, A. El-Naggar, *et al.*, “Highly accurate two-gene classifier for differentiating gastrointestinal stromal tumors and leimyosarcomas.,” *PNAS*, vol. 43, no. 102, 2007.

[7]M. Raponi, J. E. Lancet, H. Fan, *et al.*, “A 2-gene classifier for predicting response to the farnesyltransferase inhibitor tipifarnib in acute myeloid leukemia,” *Blood*, vol. 111, no. 5, pp. 2589–2596, 2008.

[8]L. Marchionni, B. Afsari, D. Geman, and J. T. Leek, “A simple and reproducible breast cancer prognostic test,” *BMC genomics*, vol. 14, no. 1, p. 336, 2013.

[9]D. Geman, B. Afsari, and D. N. A.C. Tan, “Microarray classification from several two-gene experssion comparisons,” 2008. (Winner, ICMLA Microarray Classification Algorithm Competition).

[10]B. Afsari, U. Braga-Neto, and D. Geman, “Rank discriminants for predicting phenotypes from rna expression,” *Annals of Applied Statistics*, to appear.

[11]J. Damond, *ktspair: k-Top Scoring Pairs for Microarray Classification*, 2011. R package version 1.0.

[12]J. Eddy, L. Hood, N. Price, and D. Geman, “Identifying tightly regulated and variably expressed networks by differential rank conservation,” *PLOS Computational Biology*, vol. 6, 2010.

Fig. 1. The comparisons’ votes (y-axis) vs samples (x-axis). The samples are either good (good prognosis) or bad (prognosis) for breast cancer. Truth and falsehood of the comparisons are indicated by blue and red respectively. The combination of the votes can be used as the classification rule, in this case, thresholding the number of the true comparisons by two. More explanation can be found in the vignette file and in (8)