RankGene: Identification of Diagnostic Genes Based on Expression Data

Yang Su[†] T. M. Murali[†] Vladimir Pavlovic[‡] Michael Schaffer[†] Simon Kasif[†⋆]
[†]Bioinformatics Program, Boston University, Boston MA 02215,
[‡]Department of Computer Science, Rutgers University, Piscataway, NJ 08854,
[⋆]Department of Biomedical Engineering, Boston University, Boston MA 02215.

Abstract

Summary: RankGene is a program for analyzing gene expression data and computing diagnostic genes based on their predictive power in distinguishing between different types of samples. The program integrates into one system a variety of popular ranking criteria, ranging from the traditional t-statistic to one-dimensional support vector machines. This flexibility makes RankGene a useful tool in gene expression analysis and feature selection.
Availability: http://genomics10.bu.edu/yangsu/rankgene
Contact: T. M. Murali (murali@bu.edu)

Recent research has successfully demonstrated the utility of DNA microarray-based gene expression data in cancer classification (Golub et al., 1999; Ramaswamy et al., 2001). Microarrays can identify genes that are good diagnostic indicators. Intuitively, a gene is likely to be diagnostic if its expression value in a disease state is different from its expression value in a normal state. In other words, the gene's predictive power to distinguish between different classes (e.g., normal vs. cancer) is high.

The RankGene program that we have developed ranks and selects genes based upon their ability to distinguish between various classes of samples, such as types of diseases or diseased and healthy states. The input to RankGene is a data set containing the gene expression profiles for a set of tissues or samples and the class label for each sample. For each gene, RankGene computes a value that measures the ability of the gene to distinguish between the classes. It outputs the best k genes according to this measure, where k is specified by the user. Note that our method can miss dependencies between genes that act in subtle combinations in response to disease.

There are various means of quantifying a gene's ability to distinguish between classes. RankGene supports the following eight measures:
(i) t-statistic (Golub et al., 1999): RankGene sorts the genes in decreasing order of the absolute value of the t-statistic for each gene.
(ii)–(vii) Twoing rule, information gain, gini index, max minority, sum minority, and sum of variances: these measures are widely used in the literature, have been tested experimentally, and have well-documented features and properties (Breiman et al., 1984; Murthy et al., 1994; Murthy, 1998). These measures are often called statistical impurity measures in statistical learning theory. (This notion of impurity is not related to the impurity of an RNA sample.) Each of these measures attempts to quantify the best possible class predictabilty that we can obtain by dividing the full range of expression of a given gene into two disjoint intervals corresponding to the up-regulation and the down-regulation of the gene. We predict all samples in one interval to belong to one class (e.g, normal) and all samples in the other interval to belong to the other class (e.g, cancer). Each measure quantifies the error in this prediction in a different manner. For example, the sum minority rule counts the total number of errors, assuming that the largest predicted class in each partition is correctly predicted. The information gain rule measures the reduction in class entropy resulting from the partitioning. The RankGene web page displays the formulae for each of these measures. For a given choice of measure, RankGene minimises the error over all possible thresholds that partition the gene into two

intervals.

(viii) One dimensional support vector machine (SVM): The utility of SVMs in classifying samples based on gene expression data is well-documented (Brown et al., 2000; Ramaswamy et al., 2001). We train a one-dimensional SVM on each gene's expression values. The gene's measure is the function optimised by the SVM training algorithm. Standard SVM training algorithms run in $O(n^3)$ time, where n is the number of training samples. We have developed and implemented an algorithm for training one-dimensional SVMs with linear kernels that runs in $O(n \log n)$ time (Su et al., 2003).

The software is written in C++. It has been tested on Linux and other Unix-like operating systems using the gcc compiler. The current version is aimed to be a simple screen for genes that might be interesting targets for further studies. In this version, we have primarily experimented with two-class prediction problems. Future versions of our software will attach statistical significance to each of the selected genes and will improve the robustness of the software for multi-class data.

Different measures can yield different lists of genes. In Figure 1, we compare the performance of the measures on the ALL/AML data set (Golub et al., 1999). The user can examine, compare, and collate the results of each measure in a similar manner. This ability gives RankGene the potential to be a powerful yet flexible tool in gene expression analysis.

Acknowledgments: This work was supported in part by an NSF-KDI grant.

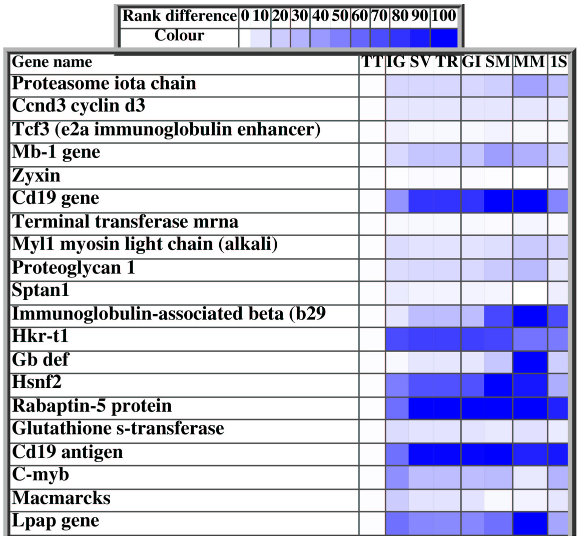

Figure 1: Comparison of the measures. The leftmost column shows the best 20 genes selected by the t-test measure. Every other column corresponds to one of the other measures. In a column for a particular measure, each entry is a colour ranging from white to blue, indicating the difference in the ranks of the gene for that measure and for the t-test. The abbreviations used are: t-test (TT), information gain (IG), sum of variances (SV), twoing rule (TR), gini index (GI), sum minority (SM), max-minority (MM), and one-dimensional SVM (1S).

References

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

M P Brown et al. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci U S A*, 97(1):262–7, 2000.

T. R. Golub et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

S. K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2(4):345–389, 1998.

S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–33, 1994.

S Ramaswamy et al. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc Natl Acad Sci U S A*, 98(26):15149–54, 2001.

Y. Su, T. M. Murali, V. Pavlovic, and S. Kasif. An efficient algorithm for training one-dimensional support vector machines. URL `http://genomics10.bu.edu/yangsu/rankgene/oned-svm.pdf`. In preparation, 2003.