Baltic J. Modern Computing, Vol. 7 (2019), No. 1, pp. 61–69
https://doi.org/10.22364/bjmc.2019.7.1.05

Cluster-Separated Classification Approach for Gene Expression Analysis

Viktar ATLIHA[1,3], Roman SERGEEV[2], Dmitrij ŠEŠOK[3]

[1] Belarussian State University, 4, Nezavisimosti ave., Minsk, Belarus
[2] United Institute of Informatics Problems NASB, 6, Surganov str., Minsk, Belarus
[3] Department of Information Technologies, Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, Saultekio al. 11, LT-10223 Vilnius, Lithuania

victor.otliga@gmail.com, roma.sergeev@gmail.com, dmitrij.sesok@vgtu.lt

Abstract. Due to dramatic progress in high-throughput sequencing technologies and widespread of microarray assays over the last decade, gene expression data has been accumulating at an accelerating pace. All this insured gene expression profiling to be extensively used as a powerful technique for phenotype classification in many biological studies. However, this is not always possible to replicate a particular experiment with various organisms or tissues to achieve sample size that will be large enough to meet the assumptions of classical statistical methods used to deliver reliable classification results. Small dataset size due to lack of sample objects can also be a problem when trying to reuse the data from public databases submitted by other researchers from their experiments. In this paper we introduce a two-step classification method for a specific task of phenotype identification, which firstly clusters data and then performs classification within each cluster. We apply this method to a real dataset for the purpose of bacterial gene-expression analysis and present its results.

Keywords: bioinformatics, gene expression, clustering, classification

1 Introduction

Depending on the research domain there are situations where the number of measured features exceeds greatly the number of available observations. This can be especially challenging for practically oriented areas such as bioinformatics and biological studies where the possibilities for experiment replication with a large sample size may be limited by the cost or technology. Most of the classical techniques that try to solve this problem are based on the introduction of the regularization term to penalize models for excessive complexity, use Akaike information criterion (AIC) or Bayesian information criterion (BIC) as a model selection criteria or apply cross-validation while estimating model parameters and performance.

Selection of the method depends greatly on the nature of the task is being solved. However, there can be experiments with only few samples available and one should make correct decision based on the whole measurements of a significant number of feature values. Straight-forward application of classical multivariate classification methods will probably show unreliable results in these conditions. Therefore this is very natural to look for small feature subsets that describe all observation with acceptable quality and sufficient for handling classifier algorithms.

Here we will present a two-stage classification method for a phenotype identification task that examines gene expression profile. This implies data clustering at the first stage to segregate more and less informative genes and then performs classification within each cluster. We evaluate this method by classifying Mycobacterium tuberculosis strains into drug-resistant versus drug-susceptible based on their gene expression profiles and separate subset of the most informative features that make the difference between these two classes. We have been focusing on M. tuberculosis whereas the problem of anti-tuberculosis drug resistance is becoming a great therapeutic challenge worldwide.

Most studies on bioinformatics analysis of tuberculosis drug resistance are aimed to identify different local features and characterize mutations in M. tuberculosis genome. For example, (Zhang et al.2013) took research among China isolates and (Korhonen et al. 2016) used data from Finland patients. There is a public database with all known single genome mutations affecting drug resistance (Sandgren et al.2009). (Farhat et al. 2013) presented a typical approach to the whole-genome analysis and its applications for determining antimicrobial resistance. (Walker et al. 2015) conducted a complex study and used a large set of sequenced genomes to propose an algorithm for identifying drug resistance based on whole-genome analysis.

Complementary to the genome sequencing, DNA microarrays are still widely used to evaluate the direct expression of genes. The most popular platforms for solving this class of problems are Affymetrix GeneChip and Illumina BeadChip microchips that employ short sequences of oligonucleotides to identify genes contained in the RNA sample. In (Fu 2006) Affymetrix system was used to determine bacterial response to drug treatments (including isoniazid) and to identify up- and down-regulated genes. This method allowed authors to detect all drug-induced genes reported in the original research and several other genes. (Fu et al. 2007) used similar approach to identify capreomycin resistance.

(Guohua 2015) described a mechanism by which the data used in our paper were obtained. He also provided a detailed analysis of the expression level of each gene using laboratory data and already known information about which genes most likely affect drug resistance. To do this, they studied how much the expression has changed after the application of the drug and the direction of this change (whether the expression level has raised or decreased). In addition, the authors of the article attempted to reveal how the expression profiles for different genes were related.

Most of the studies reviewed above are concentrating on a biological perspective without using modern methods of data analysis and machine learning. Thus, one of the goals pursued by this work is to study the transcripts from the point of the actual knowledge in the field of machine learning.

2 The proposed research model

Suppose we have a set of objects, each of which can be described using a feature vector of a fixed length. For each object we know a class it belongs to and the measurements of each feature under several conditions. The task is to identify the class of a new object based only on one feature measured under the same set of conditions.

Formally, suppose that we have n objects with m features and l different conditions. Let $c_k(x_i^{(j)})$ be the value of the j-th feature for i-th object in k-th condition. We want to build an algorithm $a(z, j)$, which would classify object based on measurements $z = \big(c_1(x^{(j)}), c_2(x^{(j)}), \ldots, c_l(x^{(j)})\big)$ for one given feature j.

In general we propose to do the following. Firstly, we will separate features into some number of clusters. This can be useful in cases when feature values differ greatly depending on the condition. And secondly, we will apply separate classification algorithms inside each cluster.

To illustrate our method we used gene expression dataset from National Center for Biotechnology Information (NCBI) project GSE53843 obtained by examining expression patterns in susceptible and resistant strains of M. tuberculosis under a number of anti-tuberculosis drugs. There were three drugs in the experiment (capreomycin, rifampicin, isoniazid) applied to each of the following bacterial strains: susceptible H37Rv, extremely drug-resistant XDR1219 and extremely drug-resistant XDR1221. We considered 3948 genes with 12 parameters for each that were levels of expression under different conditions (4 parameters for each strain, which correspond to the normal conditions with the absence of any drug and the conditions associated with the use of each of 3 drugs).

In terms of our method different strains represent the objects ($n = 3$), genes match the features ($m = 3948$) and drugs correspond to conditions ($k = 4$, one measurement in normal conditions and one for each of 3 drugs). In this task we want to separate objects into two classes: drug-sensitive (H37Rv) and drug-resistant (XDR1219, XDR1221).

More formally, let $g_i = (g_{i0}, g_{i1}, \ldots, g_{i12})$, where g_i - gene expression profile for gene i:

- g_{i0} - level of gene expression in H37Rv strain under normal conditions (without any drug used)
- g_{i1} - level of gene expression in H37Rv strain with capreomycin applied
- g_{i2} - level of gene expression in H37Rv strain with rifampicin applied
- g_{i3} - level of gene expression in H37Rv strain with isoniazid applied
- g_{i4} - level of gene expression in XDR1219 strain under normal conditions
- g_{i5} - level of gene expression in XDR1219 strain with capreomycin applied
- g_{i6} - level of gene expression in XDR1219 strain with rifampicin applied
- g_{i7} - level of gene expression in XDR1219 strain with isoniazid applied
- g_{i8} - level of gene expression in XDR1221 strain under normal conditions
- g_{i9} - level of gene expression in XDR1221 strain with capreomycin applied
- g_{i10} - level of gene expression in XDR1221 strain with rifampicin applied
- g_{i11} - level of gene expression in XDR1221 strain with isoniazid applied

In such a way we have a set of 12-dimensional vectors, each corresponding to one of 3948 genes. As it was mentioned above, the algorithm consists of two parts: clustering and further two-class classification based on the gene expression information.

2.1 Clustering

Data pre-processing was performed before the clustering step. Gene expression values were normalized against the normal gene expression levels and then logarithm was taken. The expression of each corresponding gene in H37Rv strain under normal conditions was taken as the reference value for normalization. This value was discarded during clustering. Let $t(g)$ be the transformation, as a result we obtained new 11-dimensional vectors $g'_i = t(g_i)$ that characterize gene expression level change compared to drug-sensitive strain under normal conditions:

$$t(g_i) = \left(\log \frac{g_{i1}}{g_{i0}}, \log \frac{g_{i2}}{g_{i0}}, \ldots, \log \frac{g_{i11}}{g_{i0}} \right)$$

Then we clustered 11-dimensional vectors into k groups using k-means clustering algorithm. The need for clustering is justified by the fact known from the biological findings: most of genes are uninformative for the task. Due to clustering we perform so-called "informative gene selection" to make conclusions about drug resistance of the strain from the expression levels of the "informative" genes.

2.2 Classification

In the second part all clusters were analyzed separately.

We formed 3 vectors inside each cluster for i-th gene: a_i, b_i c_i. Each vector was matched to the corresponding gene expression profile in H37Rv, XDR1219 or XDR1221 strains under different conditions respectively:

- $a_i = (a_{i0}, a_{i1}, a_{i2}, a_{i3})$
  - a_{i0} - level of gene expression in H37Rv strain in normal conditions (without any drug used)
  - a_{i1} - level of gene expression in H37Rv strain with capreomycin applied
  - a_{i2} - level of gene expression in H37Rv strain with rifampicin applied
  - a_{i3} - level of gene expression in H37Rv strain with isoniazid applied
- $b_i = (b_{i0}, b_{i1}, b_{i2}, b_{i3})$
  - b_{i0} - level of gene expression in XDR1219 strain in normal conditions (without any drug used)
  - b_{i1} - level of gene expression in XDR1219 strain with capreomycin applied
  - b_{i2} - level of gene expression in XDR1219 strain with rifampicin applied
  - b_{i3} - level of gene expression in XDR1219 strain with isoniazid applied
- $c_i = (c_{i0}, c_{i1}, c_{i2}, c_{i3})$
  - c_{i0} - level of gene expression in XDR1221 strain in normal conditions (without any drug used)
  - c_{i1} - level of gene expression in XDR1221 strain with capreomycin applied
  - c_{i2} - level of gene expression in XDR1221 strain with rifampicin applied

- c_{i3} - level of gene expression in XDR1221 strain with isoniazid applied

Vectors describing the expression of strain H37Rv were associated with class 0, which indicated that these expression levels were likely to be in a drug-sensitive strain, while the vectors describing strains XDR1219 and XDR1221 were assigned to class 1, which indicated that these expression levels were likely to be in a drug-resistant strain. Based on the whole sample obtained within each cluster, we trained logistic regression to classify the vectors of gene expression levels as belonging to a certain resistance class, which solved the task.

As a result of the classification process we calculated a vector of probabilities of belonging to each of the two classes, and if at least one of the probabilities was greater than a certain boundary, we classified the object as belonging to this class. Otherwise, if both probabilities were less than some threshold we rejected classification. Thus the rejection of classification is the special case when a classifier is not particularly sure whether strain is resistant or not. For example, if the probability of resistance is 0.55 and of sensitivity is 0.45 it is reasonably to perform a rejection because margin between probabilities is too small.

3 Computational experiments

We implemented the described above approach was using Python 2.7 programming language using major machine learning packages, such as numpy, scipy and scikit-learn. We tested it on the Dell Latitude E7440, Intel Core i7, 2.1 GHz. The algorithms done their job in approximately 2-3 seconds.

We used the k-means algorithm, introduced in (MacQueen 1967) and further developed in (Lloyd 1982) for the clustering. Logistic regression introduced in (Cox 1958) and then in (Walker et al. 1967) was used as a classification algorithm inside each cluster because of its known applicability for the genomic data (Wang 2010). Logistic regression appears to be a standard approach for binary classification problem that allows focusing not only on prediction but also on explanation. More discussion of this distinction can be found in (Shmueli et al. 2010). In current task of gene expression analysis this gives us an option to determine the influence of each separate gene expression level under considered conditions on drug-resistance status. Another reason to use logistic regression is the fact that we want a proof of concept for our approach so a particular classification algorithm is not important. This approach can be used with the other algorithms as well.

To validate the results we used a holdout validation strategy with training set of 70% and validations set of 30%. Each logistic regression classifier was learned to fit the training data and the predictions were tested on the validation set inside the cluster.

4 Results

We calculated two quality criteria: accuracy and acceptance rate. Accuracy was calculated as the proportion of correctly classified samples among those for which there

were no rejections of classification. Acceptance rate was calculated as a proportion of samples that were classified without rejection among all.

We varied the number of clusters from 1 to 8. The threshold value that affects the rejection rate of the classifier was varied from 0.5 to 0.8 in increments of 0.05. We assumed that the genes should be divided into a small number of groups based on the difference in their behavior between drug-resistant and susceptible strains under different conditions. Furthermore, in case of high threshold values the algorithm would tend to reject classification too often, which would not make practical sense.

Finally, we obtained 56 pairs of accuracy-acceptance rate values. We selected a preferable value from them taking into account that the problem is two-criteria. We wanted to choose the parameters to have the highest accuracy without drop in acceptance rate. The results are presented in Table 1. Non-dominant points are highlighted with bold.

			Number of clusters							
			1	2	3	4	5	6	7	8
Threshold	0.5	Accept rate	1	1	1	1	1	1	**1**	1
		Accuracy	0.67	0.75	0.77	0.81	0.8	0.81	**0.84**	0.83
	0.55	Accept rate	0.96	0.87	0.91	0.94	0.91	0.94	**0.89**	**0.96**
		Accuracy	0.67	0.8	0.8	0.82	0.82	0.83	**0.86**	**0.85**
	0.6	Accept rate	0.89	0.74	0.83	0.89	0.78	**0.92**	**0.81**	0.88
		Accuracy	0.68	0.89	0.81	0.83	0.89	**0.85**	**0.89**	0.86
	0.65	Accept rate	0.62	0.61	0.71	0.72	0.68	0.71	**0.73**	**0.76**
		Accuracy	0.7	0.93	0.88	0.92	0.93	0.93	**0.93**	**0.92**
	0.7	Accept rate	0.28	0.54	0.59	**0.59**	0.61	0.62	**0.64**	0.62
		Accuracy	0.78	0.94	0.93	**0.97**	0.95	0.96	**0.94**	0.96
	0.75	Accept rate	0.07	0.45	0.5	0.5	0.53	0.47	**0.55**	0.52
		Accuracy	0.83	0.97	0.96	0.97	0.97	0.97	**0.98**	0.96
	0.8	Accept rate	0.01	0.39	0.37	0.38	0.38	0.44	**0.48**	0.43
		Accuracy	0.8	0.97	0.98	0.98	0.98	0.94	**0.99**	0.97

Table 1. Quality control for different parameters

As can be seen from Table 1, all the accuracy and acceptance rate pairs that correspond to the number of clusters 7 are non-dominant, wherefrom we can assume that this is the most suitable number of clusters for our dataset.

Let's fix the minimum value of accuracy that we want to obtain when applying the algorithm. For each of these values calculate the maximum accept rate that we can get by selecting the appropriate parameters. The resulting relationship is shown on Figure 1.

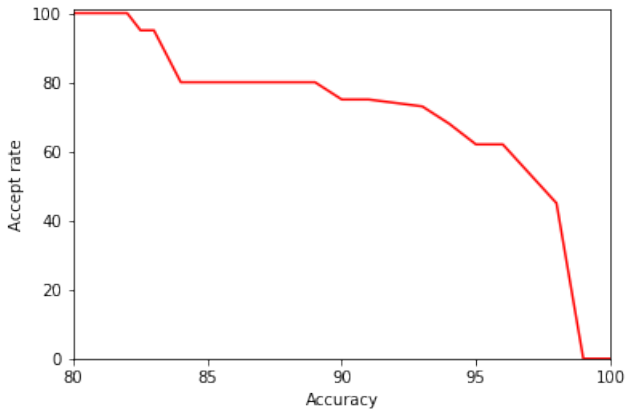

Fig. 1. Maximal accept rate level with fixed accuracy level

As we can see from the diagram if the accuracy above 90 % is required, we have a fairly small acceptance rate of about 76 %.

Similarly, lets fix the minimum accept rate that we want to obtain and calculate how the maximum achievable accuracy on it. The result is plotted on Figure 2.

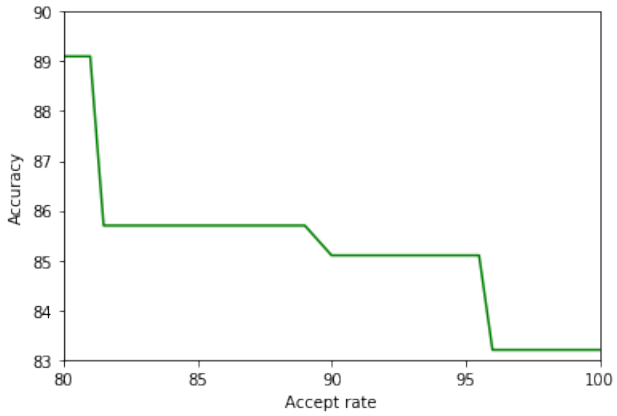

Fig. 2. Maximal accuracy level with fixed accept rate level

From this diagram we can see that if 100% acceptance rate is required (that is the total absence of classification rejections), the maximum accuracy that can be achieved is the accuracy of 84% that is a fairly good result. This is much better than a ”naive” classifier (with a probability of 1/3 that the values of gene expression correspond to their

behavior in a drug-sensitive bacteria, and a probability of 2/3 that do not correspond), which shows the acceptance rate of 100% and the accuracy of 66%. Further, it is assumed that the best result of solving the two-criteria problem is the ratio of 89%: 86% between the acceptance rate and accuracy, respectively. It is achieved with the number of clusters 7 and the value of the threshold parameter 0.55. Let's consider the values of the acceptance rate and accuracy of the classification that were obtained in each of the clusters (Table 2).

Cluster ID	Accept rate	Accuracy	Correct	Incorrect	Refused	Total
1	0.92	0.68	611	294	76	981
2	0.96	0.98	548	9	22	579
3	0.96	0.98	235	5	9	249
4	0.97	0.97	34	1	1	36
5	0.99	0.98	232	4	1	237
6	0.88	0.95	498	28	71	597
7	1	0.94	31	2	0	33
8	0.78	0.89	585	69	189	843

Table 2. Cluster characteristics

As we can see from Table 2 the accuracy in each cluster (except the first) is greater than 90%, and the value of the acceptance rate is also quite large. This means that the genes were separated by our algorithm quite qualitatively, which allowed us to build a sustainable classifier that produced good results within each cluster.

Within the first (largest) cluster much less accuracy was observed in comparison the others. This supports the ideas that clustering is an essential step to cut off non-informative genes for detection of antimicrobial drug resistance. Further research may be aimed at studying which of the clusters are most suitable for driving drug resistance.

Based on the last column of the table all samples were divided unevenly between the clusters. However, just the smallest groups show the best results both in the acceptance rate and accuracy.

5 Conclusions

The results of the experiments have showed that the suggested approach is rather useful in the task of bacterial gene expression analysis. The implemented algorithm has demonstrated fairly good results on a real dataset. At the acceptance rate of 100% it determined drug-resistance status with 84% accuracy. At the same time with the acceptance rate of 89% we could achieve 86% accuracy.

We figured out a dependency between the acceptance rate and accuracy. Various parameters could be used to reach different levels of the accuracy and acceptance rate based on the importance of them for the real-life scenario.

The proposed algorithm can be applied for binary classification problem in situations where the number of measured features exceeds greatly the number of available observations. It can be incorporated as a module of a decision support system while analyzing gene expression profiles or used under similar conditions in other problem domains, where reasonably small subset of informative features exist and application of classical multivariate classification methods seems not very rational.

References

Sandgren, A., Strong, M., Muthukrishnan, P., Weiner, B. K., Church, G. M., Murray, M. B. (2009). *Tuberculosis drug resistance mutation database*. PLoS Med., vol. 6, no. 2, pp. 132-136.

Farhat, M. R., Shapiro, B. J., Kieser, K. J., Sultana, R., Jacobson, K. R., Victor, T. C., Kaur, D. (2013). *Genomic Analysis Identifies Targets of Convergent Positive Selection in Drug Resistant Mycobacterium tuberculosis*. Nat. Genet., vol. 45, no. 10, pp. 1183-1189.

Zhang, H., Li, D., Zhao, L., Fleming, J., Lin, N., Wang, T., Zhou, Y. (2013) *Genome sequencing of 161 Mycobacterium tuberculosis isolates from China identifies genes and intergenic regions associated with drug resistance*. Nat. Genet., vol. 45, no. 10, pp. 1255-1260.

Korhonen, V., Smit, P. W., Haanper, M., Casali, N., Ruutu, P., Vasankari, T., Soini, H. (2016) *Whole genome analysis of Mycobacterium tuberculosis isolates from recurrent episodes of tuberculosis, Finland, 19952013*. Clinical Microbiology and Infection, . 22, vol. 6., pp. 549-554.

Walker, T. M., Kohl, T. A., Omar, S. V., Hedge, J., Elias, C. D. O., Bradley, P., Clifton, D. A. (2015) *Whole-genome sequencing for prediction of Mycobacterium tuberculosis drug susceptibility and resistance: a retrospective cohort study*. Lancet Infect. Dis., vol. 15, no. 10, pp. 1193-1202.

Fu, L.M. (2006) *Exploring drug action on Mycobacterium tuberculosis using affymetrix oligonucleotide genechips*. Tuberculosis, vol. 86, pp. 134-143.

Fu, L. M., Shinnick, T. M. (2007) *Genome-wide exploration of the drug action of capreomycin on Mycobacterium tuberculosis using Affymetrix oligonucleotide GeneChips*. Journal of Infections, vol. 54, pp. 277284.

Yu, G., Cui, Z., Sun, X., Peng, J., Jiang, J., Wu, W., Li, Y. (2015) *Gene expression analysis of two extensively drug-resistant tuberculosis isolates show that two-component response systems enhance drug resistance*. Tuberculosis, vol. 1-2.

Lloyd, S. (1982) *Least squares quantization in pcm*. IEEE Transactions on Information Theory, vol. 28, pp. 129137.

MacQueen, J. (1967) *Some methods for classification and analysis of multivariate observations*. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, pp. 281297.

Walker, S. H., Duncan, D. B. (1967). *Estimation of the probability of an event as a function of several independent variables*. Biometrika, vol. 54, pp. 167178.

Cox, D.R. (1958). *The regression analysis of binary sequences (with discussion)*. J Roy Stat Soc B., vol. 20, pp. 215242.

Wang, B., Wang, X. F., Howell, P., Qian, X., Huang, K., Riker, A. I., Xi, Y. (2010). *A personalized microRNA microarray normalization method using a logistic regression model*. Bioinformatics, vol. 26, pp. 228234.

Shmueli, G. (2010). *To explain or to predict?*. Statistical science, vol. 25(3), pp. 289-310.

Received May 30, 2018 , revised November 19, 2018, accepted January 29, 2019