Research paper ■

Analysis of Huntington's Disease Gene Expression Profiles with Predictive Clustering Trees

Ivica Slavkov, Sašo Džeroski, Borut Peterlin, Luca Lovrečić

Abstract. In this paper we analyzed microarray data of patients with Huntington's disease (HD). First, we preprocessed the data by using ribosomal genes expression levels as a way of normalization. After this, Predictive Clustering Trees (PCTs) were used to identify useful gene expression profiles and also to connect patient records with gene expression levels. In the end patients' pathological characteristics which created the biggest difference in gene expression levels were identified, but also genes that could serve as a way of differentiating between patients and control subjects, or presymptomatic and symptomatic patients

■ **Infor Med Slov**: 2006; 11(2): 43-51

Authors' institutions: Jožef Stefan Institute, Ljubljana, Slovenia (IS, SD), Divison of Medical Genetics, Department of Obstetrics and Gynaecology, University Medical Centre Ljubjana, Slovenia (BP, LL).

Contact person: Ivica Slavkov, Department of Knowledge Technologies, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia. email: ivica.slavkov@ijs.si.

Introduction

Huntington's disease is an autosomal dominant neurodegenerative disorder characterized by progressive motor impairment, cognitive decline, and various psychiatric symptoms, with the typical age of onset in the third to fifth decades. It is caused by the expansion of an unstable triplet repeat in *huntingtin* gene, which encodes for ubiquitously distributed huntingtin protein. Recent studies have shown that mutant huntingtin interferes with the function of widely expressed transcription factors, suggesting that gene expression may be altered in a variety of tissues, including peripheral blood. That is why it has been postulated[1] that microarray expression profiles obtained from peripheral blood samples of HD patients could be used to analyze the changes of gene expression levels caused by mutant huntingtin.

The data were obtained as a part of a study conducted by Clinical Center Ljubljana, Department of Medical Genetics.

This paper is organized as follows. First we give a brief description of the data that was available. Next we describe the preprocessing of the data, including the specific problem of re-normalizing the data between runs and filtering the important genes. The scenarios for the actual analysis of the data are further outlined. Also, a brief overview of the concept of Predictive Clustering Trees is given, as well as the software used for the analysis. At the end the results of the analysis are presented and also conclusions and plans for future work are detailed.

Description of the data

The patient records consist of three attributes: HD status (*Presymptomatic, Symptomatic, Controls*), Age (which was a continuous attribute) and Sex (*Male, Female*).

The microarray data, on the other hand, is from Slovene patients obtained from three different runs. In the first one, microarray data was obtained for 4 presymptomatic and 3 control subjects, in the second for 5 presymptomatics and 5 controls and in the third for 5 late symptomatics and 5 controls. All together, there were 27 samples. For each sample the expression levels for 54.675 probes from an Affymetrix HG.U133A 2.0 chip were measured. The expression levels were obtained by using the MAS 5.0 software.

Data preprocessing

As part of the preprocessing procedure the attribute "Age" was first discretized. Because of the typical age of onset of the disease a discretization threshold of 40 years was chosen. A bigger preprocessing effort, however, was made with the microarray data. In order to have meaningful results from the analysis a necessary condition would be to have a sufficiently large sample size. This is especially important for microarray data analysis where thousands of genes are being analyzed simultaneously and where the ratio of samples vs. attributes is always an issue. During the preprocessing of the microarray data, we tried to solve two problems. The first one was putting the data obtained from different runs into a single dataset. Despite that the ratio of genes/samples was still too big. We did a gene selection process where we decided which genes should be considered for further analysis. In the end we obtained an acceptable ratio of genes/samples, which was a tradeoff between overfitting and losing genes which might be important.

Ribosomal genes

A re-normalization of the gene expression values has to be done with the help of some stable measure. We decided to do a re-normalization with the help of the expression levels of genes, which encode ribosomal proteins. Because ribosomes are basic particles of each cell, the genes which encode them are always expressed. Therefore, their overall expression levels should be stable and also independent of the disease/control status. We used the ribosomal genes for two purposes. First we used them to check if re-normalization between runs was necessary and then we used them to re-normalize.

We marked the runs arbitrary with Run1, Run2 and Run3. The data from each run was in the format presented in Table 1.

Table 1 Data format for each run.

Gene \ Sample	S_1	S_2	...	S_n
G_1	$\exp_{1,1}$	$\exp_{1,2}$	...	$\exp_{1,n}$
G_2	$\exp_{2,1}$	$\exp_{2,2}$	...	$\exp_{2,n}$
M	...	...	...	...
G_m	$\exp_{m,1}$	$\exp_{m,2}$	...	$\exp_{m,n}$
RG_1	$\exp_{m+1,1}$	$\exp_{m+1,2}$	...	$\exp_{m+1,n}$
RG_2	$\exp_{m+2,1}$	$\exp_{m+2,2}$	...	$\exp_{m+2,n}$
M	...	...	...	...
RG_r	$\exp_{m+r,1}$	$\exp_{m+r,2}$	...	$\exp_{m+r,n}$

Explanation: In each run there were *n* samples and *m+r* genes. The set of ribosomal genes consists of the ones marked with "*RG*" and it has *r* genes.

First we calculated the values for three vectors:

$$Mean_1, Mean_2, Mean_3$$

Each element from the vector was the value of the mean of the expression levels of a ribosomal gene for the corresponding run.

$$Mean_i = \{mean.rib_{i1}, mean.rib_{i2}, \mathrm{K}\ mean.rib_{ir}\}$$

$$i = 1...3$$

where

$$mean.rib_{ij} = \frac{\sum_{l=1}^{n} \exp_{m+j,l}}{n}$$

$$j = 1...r$$

The check was made by calculating the ratios between the corresponding elements of each vector *Mean* i.e. the ratio between the means from different runs of each ribosomal gene. After this we again had three vectors containing the ratios:

$$Ratio_{ij} = \{ratio.rib_{ij1}, ratio.rib_{ij2}, \mathrm{K}\ ratio.rib_{ijr}\}$$

$$i = 1,2;\ \ j = 2,3$$

where

$$ratio.rib_{ijl} = \frac{mean.rib_{il}}{mean.rib_{jl}}$$

$$l = 1...r$$

If the values of the elements of *Ratio* are closer to "1", that means that ribosomal expression levels are comparable for Run_i and Run_j, and no re-normalization is necessary. Therefore as a numerical measure to evaluate if re-normalization between each run is needed, we used the average difference of the ratios from 1.

$$avg.diff_{ij} = \frac{\sum_{l=1}^{r} \left|1 - ratio.rib_{ijl}\right|}{r}$$

$$i = 1,2;\ \ j = 2,3$$

There was no need to re-normalize between Run1 and Run2. However, some kind of re-scaling would be necessary between Run1 and Run3, as well as between Run2 and Run3 (Table 2). Therefore, the data from Run1 and Run2 were put together (Run1-2) without re-normalization and then re-normalized with Run3.

Table 2 Average difference from "1" of the ratios of means of ribosomal genes from each run.

Average difference from "1"	Run1	Run2	Run3
Run1	0	0.113288	0.259912
Run2	0.113288	0	0.287245
Run3	0.259912	0.287245	0

Process of Re-Normalization

The algorithm of the process of re-normalization can be described as follows:

- Calculate means of expression of ribosomal genes from each run;
- Calculate ratios of means between same ribosomal genes from different runs;
- Eliminate ribosomal genes whose ratios have bigger deviation from the mean than the standard deviation;
- Use the geometric mean of the selected subset of ribosomal genes to re-normalize.

The first two steps were performed with the previously described equations. The third step was done to eliminate those ribosomal genes, which have too big difference in ratios and therefore should not be considered when determining the stable set of ribosomal genes. This step reduced the number of ribosomal genes for further consideration from 59 to 48. In the last step we took the remaining ribosomal genes and calculated their geometric mean. We then used this as a re-normalization factor.

$$rn.factor_i = \sqrt[r-k]{mean.rib_{i,1}, mean.rib_{i,2}, \mathrm{K}\ , mean.rib_{i,r-k}}$$

In the equation, *r-k* is the set of ribosomal genes which satisfy step 3 from the algorithm for re-normalization. For Run1-2 we used a re-normalization factor, $rn.factor_{12} = 8334.728$ and for Run3, $rn.factor_3 = 11327.16$. We actually divided each number (gene expression level) with its' corresponding re-normalization factor. In the end we got two new sets which could be combined directly. We calculated the new expression levels according to:

$$rn.\exp_{ijl} = \frac{\exp_{ijl}}{rn.factor_i}$$

where

$$i = 1,2;\ \ j = 1\ldots, m + r;\ \ l = 1\ldots n;$$

Selecting important genes

Although we got 27 samples all together, that was still a small number compared to 54.675 probes. That is why we wanted to filter the genes further. First we eliminated the "background noise" genes. Only the genes with expression level bigger then 100 in at least one sample in each run separately were included in further analysis. With this we ensured that the data obtained from a single run was not just a background noise. Further testing was done on the full sets. In order to identify genes which were differentially expressed in HD patients compared to controls a significance test (student t-test) between patients and controls was performed. Only genes with $p<0.05$ were considered. Then the means of gene expression levels of HD patients and controls were calculated separately. We compared the ratio of these means and by using cut-off values of <0.6 and >1.8 we decided upon the final set of genes that were potentially significant for further analysis.[1]

At the end of the preprocessing phase we got a dataset of 27 patients and 109 probes (genes).

Data analysis using predictive clustering trees

Considering the previous description of the data there were two major tasks. First we wanted to see the connection patient records-microarray data and then microarray data-patient records. We used decision trees or the more general Predictive Clustering Trees (PCTs) as a way of connecting the data. The software that was used is a generic system for constructing decision trees- Clus. Before proceeding to details about the analysis a brief description of PCTs and Clus is given.

Predictive Clustering Trees (PCTs)

Decision trees are usually considered for classification purposes. Each tree consists of three elements: internal nodes, branches and leaves. The internal nodes are labeled with some attribute (variable name) and each branch is labeled with a predicate that can be applied to the attribute associated with the parent node. The leaves however, are labeled with a class. Following the branches from the root to a leaf gives sufficient conditions for classification (Figure 1).

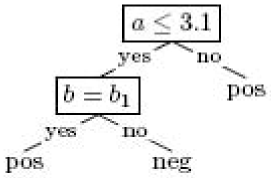

Figure 1 A typical classification tree with classes "pos" and "neg".

An alternate view of decision trees is that they correspond to the concept of hierarchical clustering.[2,3] Each node (and leaf) corresponds to a cluster and the tree as a whole represents a kind of taxonomy or hierarchy. Thus we can use the concept of TDIDT (Top-Down Induction of Decision Trees) for inducing clustering trees. We assume that two types of functions exist. A prototype function, which is used to get the best description of the members of a cluster, and a distance function for measuring the distance between prototypes and also between members of the cluster and the prototype. When inducing the clustering tree the TDIDT algorithm uses as a heuristic the minimization of intra-cluster variance (and maximization of inter-cluster variance). The minimization of the intra-cluster variance means minimizing the average distance between the members of the cluster and the prototype, which describes it. Maximization of the inter-cluster variance maximizes the distance between the prototypes. At the end we get a clustering tree in which the top-level node corresponds to one cluster containing all of the data, which is recursively partitioned into smaller clusters while moving down the tree. The leaves of the clustering tree are clusters, but they also store information about the cluster prototype. Because in essence the prototype describes the cluster, it can also be considered as a prediction of the values in that cluster with a certain amount of error.

The Clus system

The system that was used for data analysis is called Clus. It is a generic system for constructing decision trees. It can be used for constructing classification trees, for predicting symbolic attributes, as well as for regression trees for numeric values prediction. Sometimes it is also useful for predicting several attributes at once so multi-objective trees can also be constructed (Figure 2).

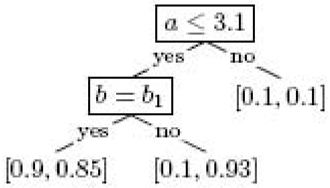

Figure 2 Multi-objective regression tree predicting values for two numeric attributes.

Clus uses a standard recursive top-down induction algorithm to construct the decision trees similar to that of C4.5 and Cart.

We use Clus by executing it in a "beam-search" mode. When run in this mode, Clus considers the set of *k* best trees (the beam) found so far. This beam when initialized has one tree that consists of only one leaf node. During each iteration of the beam-search, the beam is modified by constructing all refinements of all trees in the beam. A refinement is obtained by replacing a leaf by a new internal node. This is repeated for each possible test that can be used in the internal node. If a refinement is better than the worst tree in the beam, then this tree is replaced by the new refinement. The score for sorting the trees in the beam can be based on accuracy or on entropy. The execution in beam-search mode ends if no better refinements are found and the beam of the *k* best trees is returned.

Simulation of IC-clustering

The first scenario of analysis was the simulation of the Itemset Constrained clustering (IC-clustering)[4] as a way of connecting patient records-microarray data.

The idea of simulating IC clustering is the following: We have the patient attributes (Age>40, SexMale…) or combination of attributes (Symptomatic&Age>40, Age>40&SexMale…) which divide the patients (samples) into two groups- one group that has those characteristics (e.g. Age>40) and another one which does not (e.g. Age<40). We want to determine which attribute (or combination of attributes) creates groups whose members are most similar to each other in terms of their gene expression levels.

The steps for this scenario for analysis are:

- Find frequent itemsets from the patient records (attributes);
- Use them as patient features;
- Create PCT stubs with patients' features as constraints.

At the beginning we wanted to see which patient attributes appeared together most often- like "Symptomatic_Age>40". After finding the frequent itemsets we used them as features. This means that if the patient (sample) is symptomatic and has more than 40 years of age, we assign the Symptomatic and Age>40 attribute but also the Symptomatic_Age>40 as a separate attribute. We use Clus in a beam search mode to construct PCT stubs (Figure 3) which are trees which contain only one internal node. The node contains the patient feature (the patient features are the descriptive attributes *D*) and the leaves just contain the information of the cluster size.

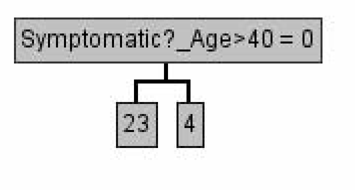

Figure 3 PCT stub which is output from Clus.

Determining useful gene expression profiles

When talking about useful gene expression profiles we consider genes or set of genes which might prove useful for distinguishing two things: if a subject has HD or is healthy, but also to determine the progress of the disease of HD patients. First we tried to construct regular classification trees. The class we tried to predict was "Huntington" with class attributes *HD* and *H* (healthy). Then we used as a class "Stage" with three class values: *P* (presymtpomatic), *S* (symptomatic) and C (controls). For determining the accuracy of the decision trees we used a leave-one-out validation. However the accuracy was low in both cases. In the case of the class "Huntington" the accuracy was around 51% and in the case of Stage it was even lower around 44% (Table 3).

Therefore we tried to improve the accuracy of the models by constructing PCTs. The PCTs were more specifically multi-objective decision trees that used the two classes “Huntington” and “Stage” simultaneously (Figure 4). We persumed that this would help the algorithm to do a more informed decision when constructing the trees.

Table 3 Comparison of accuracy for different types of decision trees

Type of model	Class	Class values	Accuracy
Classification tree	Huntington	{HD,H}	51%
Classification tree	Stage	{P,S,C}	44%
Multi-objective classification tree	Huntington	{HD,H}	74%
	Stage	{P,S,C}	74%

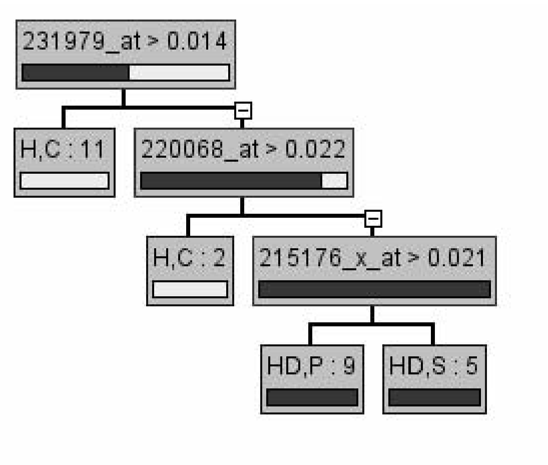

Figure 4 Multi-objective decision tree with genes at the decision nodes.

Table 4 The relationship between the two classes “Huntington” and “Stage”.

Stage\Huntington	HD=true, H=false		HD=false, H=true
P	true	false	false
S	false	true	false
C	false		true

The cross-validation showed that the accuracy rose to 74% for both classes. The reason why multi-objective decision trees performed better was because the two classes were correlated (Table 4).

In essence, we have put a constraint on the algorithm when trying to separate Presymptomatic, Symptomatic and Controls to prefer putting Presymtpomatic and Symptomatic in one cluster and Controls in the other. Furthermore, we used Clus in a beam search mode in order to see which are the best multi-objective trees that can be constructed. We found out that the top 203 trees have the same value for the intra-cluster variance. Each of these trees had three genes in its internal nodes.

Results from the simulation of IC-clustering

The results from the simulation of IC-clustering (Table 5) were somewhat expected. The attribute, which created clusters that had the minimal values for the intra-cluster variance was “Symptomatic”. This confirms the results from a previous study of HD[1] that the biggest differences of gene expression levels were detected in HD patients that were symptomatic with respect to the presymptomatic and control subjects.

Table 5 Results from the simulation of IC-clustering.

Cluster rank	Cluster description
1	Symptomatic
2	Control
3	HD
4	Symptomatic_Age>40
5	HD_Age>40
…	…

The clusters “Control” and “HD” had the same value for the intra-cluster variance and they were also expected results because they show that the clusters are less compact if they contain presymptomatic and symptomatic patients together i.e. HD patients. Symptomatic and Age>40 stresses the significance of the age of onset of the disease connected to the appearing of the symptoms of HD.

Results obtained with Predicitve Clustering Trees

As mentioned previously the 203 models with the same intra-cluster variance were obtained from the beam search that was conducted with Clus. They were in essence the 203 models of multi-objective classification trees, which were the best predicitive models of the data and for which the same accuracy assessment from the leave-one-out validation applied. The probes which were part of these models were identified and ranked according to their number of appearance in the models. In this way we identified altogether 39 probes. To see if any of the probes (genes) identified as significant were meaningful in biological context we searched for their function from the gene ontology database.[5]

For seventeen of them little or nothing is known and cannot be matched to a known gene of biological function.

Probes "1556462_a_at", "226736_at", "205101_at","1553983_at"are involved in processes related to transcription - transcriptional activator activity, transcription factor activity, transcription corepressor activity, pyrimidine metabolism, dTDP and dTTP biosynthesis, DNA metabolism. It has been previously shown that disturbed transcriptional activities are characteristic for HD.[6,7] Therefore our finding that expression of these transcription-related genes is different in above described groups is even more suggestive that the transcription in HD is disturbed.

Also, inappropriate protein functioning, synthesis and degradation has been documented in HD.[8] Probes "1558699_a_at", "204581_at", "240254_at", "1555181_a_at", "201041_s_at", "228055_at", 228056_at", "204410_at" are all involved in different processes related to proper protein functioning, for example proteolysis and peptidase activity, protein modification, protein binding, protein amino acid glycosylation, protein biosynthesis, translational initiation or regulation of translation.

Probe "221478_at" is predicted to have a role in induction and positive regulation of apoptosis, which is another process that is balanced out in HD.[9]

Some of the remaining probes are involved in immune response ("209374_s_at","212671_s_at","222934_s_at"), intracellular signaling cascade ("239533_at","204484_at") and some other processes, which are yet to be elucidated in the view of HD.

Conclusion and further work

In this study a lot of effort was made in the preprocessing phase and in finding a way to do a proper normalization of the gene expression data. Also, reducing the ratio of genes/samples was addressed and statistical filtering of the important genes was performed. As part of the analysis process of the microarray data an attempt was made to demonstrate the uses of PCTs for two purposes: as a way to connect the patient records with microarray data and also as a way to identify important gene expression profiles. Some of the genes that were identified could prove useful as biomarkers for Huntington's disease, but further work and biological insights are needed in order to be able to make any kind of a significant claim and true validation of the results. This would include validation of the results by repeating the analysis on other HD microarray data sets and also by comparing results from different types of analysis and previous studies.

References

1. Borovecki et al.: Genome-wide expression profiling of human blood reveals biomarkers for

Huntington's disease. In *Proceedings of the National Academy of Sciences of the USA*, August 2 2002, vol. 102., no 31, p 11023-11028

2. H. Blockeel. *Top-down induction of first order logical decision trees*. PhD thesis, Department of Computer Science, Khatolieke Universiteit Leuven, 1998.
3. H. Blockeel, L. De Raedt, and J. Ramon. Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning*, pages 55–63, 1998.
4. Sese Jun, Yukinori Kurokawa, Kikuya Kato, Morito Monden and Shinichi Morishita. Constrained Clusters of Gene Expression Profiles with Pathological Features. *Bioinformatics* vol. 20 issue 17 Oxford University Press 2004
5. Web Page: http://www.geneontology.org.
6. Dunah AW, Jeong H, Griffin A et al. Sp1 and TAFII130 transcriptional activity disrupted in early huntington's disease. *Science*, 2002; 296: 2238-43.
7. Sugars KL, Rubinsztein DC. Transcriptional abnormalities in Huntington disease. *Trends in Genetics*, 2003; 19: 233-238.
8. Harjes P, Wanker EE. The hunt for huntingtin function: interaction partners tell many different stories. *Trends Biochem Sci.* 2003; 28 (8): 425-33.
9. Sawa A, Wiegand GW, Cooper J et al. Increased apoptosis of Huntington disease lymphoblasts associated with repeat length-dependent mitochondrial depolarization. *Nat Med* 1999; 10: 1194-1198.