Predicting the Effects of Gene Deletion

David S. Vogel, M.S.
A.I. Insight, Inc.
602 Courtland St., Suite 400
Orlando, FL 32804
dvogel1@cfl.rr.com

Randy C. Axelrod, M.D.
Sentara Healthcare
6015 Poplar Hall Drive
Norfolk, VA 23502
rcaxelro@sentara.com

ABSTRACT

In this paper, we describe techniques that can be used to predict the effects of gene deletion. We will focus mainly on the creation of predictive variables, and then briefly discuss different modeling techniques that have been used successfully on this data.

Keywords

KDD Cup, KDD, gene, deletion, prediction, model, interaction, decision tree, essential, text mining.

1. INTRODUCTION

The 2002 KDD Cup was an exercise in our ability to predict the effects of gene deletion. Given to the contestants were a series of relational tables as well as numerous abstracts discussing many of the genes. By using text mining algorithms and relational algorithms, we built a series of 1000 variables that could potentially predict the narrow and broad classes of the problem. Because the data was sparse, we had to come up with a way to select the best variables, combine those variables into group variables, and apply a low complexity model to avoid over-fitting.

2. INDEPENDENT VARIABLE CREATION

The tables included information on three specific properties for the genes. The properties included location of the gene action in relation to the cell structure (e.g. cell membrane or nucleus), the protein class of the protein produced or activated by the gene (e.g. actins or cyclins), and the function of the protein produced as it related to cellular function (e.g. metabolic or energy related). The properties for each gene lay within multilevel architecture or hierarchy. These hierarchical tables also were available. In addition, tables of gene-to-gene interactions and gene aliases were given with dependent variables (the narrow and broad positive classes) for the training set. Finally, abstracts and tables relating the gene aliases were provided for mining. We used text-mining algorithms to convert this text into a set of numeric variables that would likely be predictive of the two classes.

2.1 Gene Characteristics

The creation of gene characteristic variables was conducted in such a fashion as to allow the processing between the gene specific characteristics and the hierarchal structure of the three gene characteristics. Several variables would be created for each gene and its action location characteristics. These variables would reflect the specific location as well as any sub grouping characteristics the gene contained as a result of the hierarchy of the location table. All variable and hierarchal levels were designed to be binary for each potential level of location, protein and protein function. An example of this variable set for gene YOR113W and protein class would equal 1 for Proteases, Transcription factors, Zinc-coordinating DNA-binding domains, and Cys2His2 zinc-finger and zero for all other groups and subgroups of protein classes.

2.2 Gene Name Variables

More variable creation was derived by concatenating the naming nomenclature of the alias names to derive several variables from the text nomenclature itself. Examples for gene YOR113W would be Y, YOR, OR, 1, 13, 113, and W. This was done to identify any naming nomenclature characteristics that could offer predictive values. High frequency gene aliases also were examined in this fashion.

2.3 Text Mining Variables

Most of the key variables did not come from the data tables that were given. The key variables were generated based on a text-mining algorithm searching a database of 15,234 abstracts discussing these genes. Our first step was to browse through several articles containing narrow and broad class genes, and to compile a list of words that intuitively appeared like they could be predictive key words. The final list contained 23 words: change, chromosome, delete, direct, elusive, ethanol, FK506, glucose, Hst1, ime2, inhibit, interact, molecular, ndt80, promoter, radioactive, rapamycin, repress, sensitive, Set3C, signal transduction, transcription, and tumor. We also equated plurals, past tense, and other forms of the same word to be interpreted in the same way. For each of these key words, we generated 3 variables, as we were not sure which would be more predictive: 1) A count of how many times a key word appeared in the same article as a gene, 2) A count of how many times a key word appeared within 100 characters of the gene, and 3) A count of how many times a key word appeared within 60 characters of the gene. As it turned out, there was no general rule of which of the 3 were more predictive, as it varied from one key word to another. The strongest predictors out of the entire data set were based on the key words: complex, mutation, interact, and essential. Of the 630 genes that appeared in an abstract containing the word "interact", 21 were narrow class genes, and 11 were broad class genes. This single variable narrows down more than half of the narrow class genes in a little more than one fifth of the data set. It should also be noted that sorting the test set by this variable alone yields an ROC area of 0.6516, enough to rank 3rd on the narrow positive class in the KDD Cup.

2.4 Interaction Variables

The first idea is obvious: Check to see which genes interact with the target genes. This idea quickly is disregarded as one finds that not a single gene interacts with more than one gene in the predicted classes, thus wiping out the possibility of any statistical significance for the predictability of an interaction with a single gene. Additionally, it can also be seen that about one out of four genes in the predicted classes have any interactions at all, so at best the presence of interactions can be one fourth of the solution. The total number of interactions was a strong predictor. 5.5% of the 128 genes with 10 or more interactions were in the narrow class (compared to 1.1% of the remaining genes). Furthermore, by evaluating various groups of genes interacting with the target genes (primarily the narrow partition), some strong patterns were observed. How did we determine which groups of genes to evaluate for interactions? We used all of our 500 variables as groups and generated 500 interaction variables, each representing the number of interactions with genes having that variable's value greater than 0. The most predictive interaction was evaluating interaction with essential genes. Out of 260 genes interacting with this group, 12 were targets in the narrow class (out of only 15 total narrow class genes with any interactions at all).

3. MODELING TECHNIQUES

The biggest problem in modeling this data was the sparseness of the data. There were very low frequencies of target values, and none of the variables were extremely predictive. This required a model of extremely low complexity to avoid over-fitting. Furthermore, we felt that one could not split out a validation/test set since further dilution of the frequencies would both weaken the model significantly and give an unreliable test set result. Given more time, re-sampling methods could have been used to test the reliability of our models. However, our time limitations led us to decide to design a single model, and try as best we could to avoid over-fitting. To demonstrate how quickly over-fitting could occur with this data, we ran linear regression on the best 6 variables. For the narrow class, the ROC area was 0.6350 (on the test set), which is less than the ROC area when sorting by the single best variable. In section 3.1, we discuss the techniques actually used for our submission in the KDD Cup. In section 3.2, we discuss ideas for improvement after taking away the time constraints of the competition.

3.1 Decision Tree and results

First, we sorted our variables by correlation to the dependent variable, and chose 0.08 as a cutoff, since that included 54 variables, which was more than enough to work with. Within that group, we evaluated subsets of the dataset where each of the variables was greater than 0, and also the subset where the value was greater than 1. Within each of these subsets, we decided upon which ones had significantly high percentages of the dependent value. 13 variable ranges appeared the best. We then created a variable that was a count of these ranges, and named it "GoodVar1". Of the 141 genes that had a "GoodVar1" value of 3 or greater, 14 were in the narrow class. This became the first key split of our decision tree, which we further split based on combinations of variables that appeared to show the highest predictability. In the subset that did not have a "GoodVar1" value of 3 or more, we ran a similar variable selection process, and combined a second set of 25 variables into a variable named "GoodVar2." Variables with a "GoodVar2" value of 5 or more proved to be highly predictive, so that became our second key decision cut point. In further decision tree splits, we did not use an automated algorithm, due to the sparseness of the data relative to the complexity of an automated decision tree. Each cut point was manually decided upon based on looking at statistics of the predictability of variables within the decision tree node, and then checking to see which ones were backed up by predictability over the entire data set. If we had not done this check, there would have been no statistically significant predictors within our nodes, due to the tendency of decision trees to make data progressively more sparse. As we found ourselves very close to the deadline for submitting our entry, we did not create combination variables for the broad class. We simply used "GoodVar1", "GoodVar2", and the highly correlated variables to the broad class. We then used the same decision tree technique that we used for the narrow class.

3.2 Retrospective Improvements

For purposes of the KDD Cup, time constraints prevented many teams from utilizing the best possible model, but rather a quick modeling technique. For purposes of this paper, we made a few modeling improvements and applied a second model (narrow partition only) to the test set to give a better idea of what the potential of the data set could be. Under the time constraints, variables with the highest correlation to the dependent variables were the ones considered, as this is a quick and easy calculation. However, there are better techniques for variable selection that could have been used that would be less influenced by random patterns. In our case, we examined subsets of the data where independent variables were greater than 0, and greater than 1. Within these subsets, we defined a measure that evaluated the size of the predictive value minus a penalty for potential noise. This measure we defined as $N_{NP} - E(N_{NP}) - SD[E(N_{NP})]$. N_{NP} represents the number of occurrences of the narrow partition within the subset. $E(N_{NP})$ represents the expected number of occurrences, based on the frequency within the entire dataset. $SD[E(N_{NP})]$ is the standard deviation of this expected number of occurrences. When remodeling using this measure of predictability, we also created 4 groups of variables similar to the principle of "GoodVar1" and "GoodVar2" discussed in the previous section. We applied this model, and obtained a test set result of 0.70, a large improvement from our previous result.

4. DISCUSSION

Due to the sparse data, and lack of solid variables, it was important to have a very low complexity model. We accomplished this by combining several predictive variables into grouped variables. We also kept the model as simple as possible by manually partitioning a decision tree. We scraped for more data by forfeiting a test set. We also improved the power of our variables by creating the 500 interaction variables (that is, gene interactions, not data mining interactions) .

5. ACKNOWLEDGMENTS

We acknowledge A.I. Insight, Inc. and MEDai, Inc. for the use of their proprietary predictive modeling technology, MITCH. (Multiple Intelligent Tasking Computer Heuristics)

6. REFERENCES

[1] *Saccharomyces* Genome Deletion Project web page, the list of essential ORFs link: http://www-sequence.stanford.edu /group/yeast_deletion_project/Essential_ORFs.txt

[2] Functional Profiling of the Saccharomyces cerevisiae Genome. Nature 418 : 387-391 (2002).

[3] Winzeler, E., Shoemaker, D., Astromoff, A. Liang, H., et al. Functional Characterization of the Saccharomyces cerevisiae Genome by Gene Deletion and Parallel Analysis. Science. 285, 901-906. (1999).

[4] Shoemaker, D., Lashkari, D.A., Morris, D., Mittmann, M. & Davis, R.W. Quantitative phenotypic analysis of yeast deletion mutants using a highly parallel molecular bar-coding strategy. Nature Genetics. 14, 450-456 (1996).

About the authors:

David Vogel studied Mathematics at M.I.T. and earned a M.S. in Scientific Computing at N.Y.U. For his thesis, he invented a new technique for calculating radiation exposure, which was several times quicker and more accurate than the industry standard. After leaving the Nuclear Engineering industry in 1998, Mr. Vogel found himself at A.I. Insight, where he is now the Senior Scientist. Having been unsatisfied with the mainstream predictive modeling software packages that are available to be purchased, Mr. Vogel led the way at A.I. Insight in the development of MITCH (Multiple Intelligent Tasking Computer Heuristics). This software has been applied toward the medical industry, where it has exceeded the level of accuracy that has been published in medical journals as the "Theoretical Maximum" predictive accuracy. Mr. Vogel has also incorporated MITCH into the creation of a stock model, now being licensed for the purpose of managing an investment fund. Mr. Vogel has submitted entries for the KDD Cup in 2001 and 2002, and has earned honorable mentions on both occasions.

Randy C. Axelrod, M.D. is the Vice President and Executive Medical Director for Sentara Healthcare in Norfolk, Virginia. He received his BS from Tulane University and his MD from the University of Cincinnati. Dr. Axelrod has been with Sentara for nearly seven years focusing on clinical information and health care outcomes research and improvement for the Hampton Roads community. His efforts have focused on the integration of A.I. technology into health care delivery on the provider side as well as the payer side. His improvements in clinical outcomes and innovations have received many national awards. In 2001, Dr. Axelrod received the Innovator's Award from the Health Insurance Association of America and National Underwriter for his work with artificial intelligence in the health insurance industry.