Classifying Genes into Gene Ontology Categories Using Text-Based Supervised Learning Methods

Frédéric Tingaud, Tomonori Izumitani, Hirotoshi Taira and Eisaku Maeda

NTT Communication Science Laboratories
Nippon Telegraph and Telephone Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan
{tingaud, izumi, taira, maeda}@cslab.kecl.ntt.co.jp

keywords: Gene Ontology, Annotation, NLP, Supervised learning, k-Nearest neighbors, SVM

Introduction

The largest amounts of information characteristic of the post-genomic era make the annotation of genes more and more important. Some projects have been launched to perform high-scale annotation using controlled vocabularies, such as the Gene Ontology Project[4]. However, these are constructed manually, which is very time consuming. Moreover, the structure of vocabularies evolves, and maintenance is thus very demanding. Inferring annotation automatically can greatly simplify this task. Raychaudhuri et al.[2] proposed such a method.

Their method classifies documents about genes using supervised learning and statistical Natural Language Processing (NLP) techniques. The gene category is then deduced from the category of the documents using a voting scheme. However, this method only determines the ranking of the categories, not the applicability. Additionally, the authors did not use information about already annotated genes, which can be very informative.

We propose a method to classify genes into multiple gene ontology (GO) categories using supervised learning and NLP. We use the categories of annotated genes to label the training data.

Result and Discussion

We represented each gene as a vector of weighted term frequencies calculated from the set of documents referring to it. The weighting is determined by the tf×idf[3], which is commonly used in NLP. We classified genes using two document classifiers. The Support Vector Machines (SVM)[5], which determine the hyperplane maximizing the margin between two groups, and the k-Nearest Neighbors (k-NN), which use the nearest known items to infer the class of a tested item.

As a gene can be classified into more than one category, we used binary classifiers. There are as many classifiers as categories, and each classifier determines if the gene fits a category or not.

We applied our method to the classification of yeast genes registered in the Saccharomyces Gene Database (SGD; http://genome-www.stanford.edu/Saccharomyces/)[1]. Most of these genes are already annotated with GO codes, and have references to PubMed abstracts.

We selected genes whose GO category is one of the 11 categories also used by Raychaudhuri et al.[2]: “Biogenesis”, ”Cell Cycle”, “Homeostasis”, “Intracellular”, “Invasive”, “Meiosis”, “Metabolism”, “Signal Transduction”, “Sporulation”, “Stress Response” and “Transport” (“Cell Fusion” is now deprecated). Since the GO is a hierarchical set of terms, we considered the “children” as part of the “parent” category. We ignored genes with less than three or more than 25 abstracts.

We downloaded the selected abstracts and represented genes as vectors of terms, inserting all the terms present for more than five genes and less than 1,500. Finally, we obtained 1,969 vectors of 11,701 dimensions.

Table 1. Five-fold cross validation of GO categories prediction

GO categories (num of genes)		SVM Poly1	SVM Poly2	k-NN k = 1	k-NN k = 5
Biogenesis (562)	P	0.7811	0.8210	0.7616	0.7523
	R	0.7379	0.6808	0.7593	0.7351
Cell Cycle (226)	P	0.7451	0.7220	0.7052	0.7714
	R	0.7043	0.6050	0.7291	0.6311
Homeostasis (55)	P	0.7125	0.6421	0.5861	0.6706
	R	0.5264	0.4799	0.5058	0.4183
Intracellular (188)	P	0.7798	0.7244	0.7188	0.7466
	R	0.6668	0.6642	0.7406	0.6711
Invasive (7)	P	0.6000	0.6000	0.5	0
	R	0.5000	0.5000	0.6	0
Meiosis (43)	P	0.5463	0.6040	0.4564	0.8191
	R	0.5400	0.4667	0.5311	0.4822
Metabolism (1527)	P	0.9069	0.8913	0.9153	0.8929
	R	0.9313	0.9338	0.9195	0.9281
Signal Transduction (66)	P	0.7832	0.7171	0.7689	0.7617
	R	0.7038	0.6271	0.6786	0.5067
Sporulation (33)	P	0.5429	0.4567	0.3333	0.6333
	R	0.4111	0.3444	0.3806	0.2389
Stress Response (111)	P	0.5778	0.5043	0.6002	0.5738
	R	0.4414	0.3608	0.5316	0.2714
Transport (414)	P	0.8158	0.7957	0.7542	0.7775
	R	0.7722	0.6799	0.7959	0.7358

We evaluated the performance of the proposed method by five-fold cross validation. Table 1 shows the results for the different classifiers. For each category, we calculated the Precision (P) and the Recall (R). We used two different kernels for the SVM classifier: a linear kernel (Poly1) and a 2nd-degree polynomial kernel (Poly2). In this study, the linear kernel tends to produce better results. For the k-NN, we calculated distances beetween the vectors, using the cosine distance measure. Table 1 shows the results for k = 1 and k = 5.

From the results of SVM (Poly1) and k-NN, the four categories represented by more than 200 genes have both precision and recall higher than 70%. On the other side, the two categories with less than 50 genes correspond to low recall and precision. It therefore seems that the size of the training set is very important for the quality of the results. k-NN gives better results than SVM for classes with lots of genes, whereas SVM is more robust in the case of classes with few genes.

These results cannot be compared directly with Raychaudhuri et al.'s results, because when they developed their method, the SGD database contained fewer genes than it does now, so they tested it on a smaller sample. Additionally, their method only outputs rankings, so their method cannot be evaluated by the same recall and precision metrics as ours. They used two other ways to evaluate the efficiency of their method. For each category, the "Exact Match" is the percentage of correctly assigned genes. The "Partial Match" corresponds to the "Exact Match" plus the percentage of genes assigned to any other correct categories, in the case of multiple correct categories. The "Exact Match" is stricter than our method, while the "Partial Match" is less strict. Our method produced better results than the "Exact Match" for nine categories with SVM (Poly1), ten with k-NN/k=1 and eight with k-NN/k=5. Furthermore, SVM, k-NN/k=1 and k-NN/k=5 gave better results than their "Partial Match" for five, six and four categories respectively.

With SVM (Poly1), for 91.73% of the genes, at least one of the correct categories was predicted. For 64.55%, all of the correct categories were predicted exactly. k-NN/k=1 predicts 92.03% and 65.31%, and k-NN/k=5 predicts 90.91% and 62.37%, respectively for all correct categories and for any correct category.

These results suggest that the method can efficiently help curators annotate genes. Further development could use our method on species other than yeast and for the predictive annotation of unknown genes.

References

[1] S. S. Dwight, M. A. Harris, K. Dolinski, C. A. Ball, G. Binkley, K. R. Christie, D. G. Fisk, L. Issel-Tarver, M. Schroeder, G. Sherlock, A. Sethuraman, D. Weng, S. Botstein, and J. M. Cherry. Saccharomyces Genome Database (SGD) provides secondary gene annotation using the Gene Ontology (GO). *Nucleic Acids Res.*, 30(1):69–72, 2002.

[2] S. Raychaudhuri, J. T. Chang, P. D. Sutphin, and R. B. Altman. Associating genes with Gene Ontology codes using a maximum entropy analysis of biomedical literature. *Genome Res.*, 12(1):203–214, January 2002.

[3] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[4] The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[5] V. N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag, New York, 1999.