Team Project

Function prediction using CDD

Members:

李新炜　**11307110147**

修子迪

李柏逸

Abstract

A variety of functional genomics experimental techniques are available, from classic methods such as affinity precipitation to advanced high-throughput techniques such as gene expression microarrays, but those techniques are so exhaustive and inefficient that computational methods to analysis data are needed. Thanks to the DNA sequencing technology, genomic sequencing is no longer a novelty. Characterizing gene function, however, is one of the major challenging tasks in post-genomic era. To address this challenge, we use the Conserved Domain Database (CDD)[1] to predict some unknown yeast genes function by several computational methods. The methods we use including (1) simple-score method, (2) Fisher's exact test[2] method, (3) complex-score method and (4) naive Bayesian method are based on novel assessment for the relationship with (1) CD (2) SG and (3) GO[3,4].

Introduction

CDD

NCBI's Conserved Domain Database (CDD) is a resource for the annotation of protein sequences with the location of conserved domain footprints, and functional sites inferred from these footprints. CDD includes manually curated domain models that make use of protein 3D structure to refine domain models and provide insights into sequence/structure/function relationships. Manually curated models are organized hierarchically if they describe domain families that are clearly related by common descent. As CDD also imports domain family models from a variety of external sources, it is a partially redundant collection. To simplify protein annotation, redundant models and models describing homologous families are clustered into superfamilies. By default, domain footprints are annotated with the corresponding superfamily designation, on top of which specific annotation may indicate high-confidence assignment of family membership. Pre-computed domain annotation is available for proteins in the Entrez/Protein dataset, and a novel interface, Batch CD-Search, allows the computation and download of annotation for large sets of protein queries.

GO

The GO project provides ontologies to describe attributes of gene products in three non-overlapping domains of molecular biology. Within each ontology, terms have free text definitions and stable unique identifiers. The vocabularies are structured in a classification that supports `is-a' and `part-of' relation- ships. The scope and structure of the GO vocabularies are described in more detail in references (5±7). In the current research environment, where new genome sequences are being rapidly

generated, and where comparative genome analysis requires the integration of data from multiple sources, it is especially germane to provide rigorous ontologies that can be shared by the community.

Method

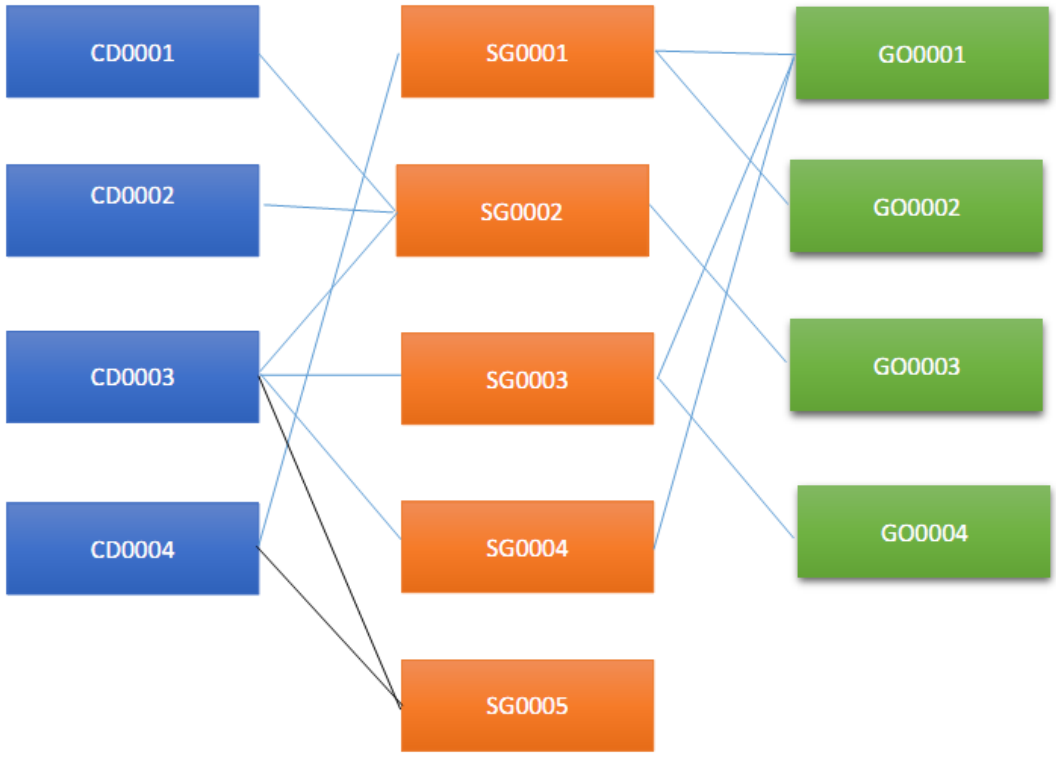


Figure 1: A sample of CD~SG~GO relationship

Simple-score method

In our project, we have the e-value（0~10） between the CD and SG. And we transfer the e-value to C-score.

C-score=-ln（e_value+10^-99）+2.4

In Figure 1, for example, there are the relationship with CD, SG and GO. The relationship between CD0003 and GO0001 C-score is added by C-scores between CD0003 / SG0003 and CD0004 / SG0004, so that we can get every C-score between CD and GO.

To predict the unknown relationship between SG0005 and GO, we can calculate the Final-score by known relationship between SG0005 and CD.

GO0001 Final-score = C -score（CD0003 / Go0001）+
C -score（CD0004 / Go0001）

GO0002 Final-score = C -score（CD0004 / Go0002）

GO0003 Final-score = C -score（CD0003 / Go0003）

GO0004 Final-score = C -score（CD0003 / Go0003）

Fisher's exact test method

We calculate the number of SGs use with/without CD and with/without Go. Use data in figure we can get the follow table describing the relationship of CD0003 and GO0001.

SG	With GO0001	Without GO0001
With CD0003	2	1
Without CD0003	1	0

Than we perform Fisher's exact test on every SG, and the lower P value, the stronger correlation between CD and GO.

To predict the unknown relationship between SG0005 and GO, we can calculate the Final-score by known relationship between SG0005 and CD.

Final-score =

-ln（P（CD0003 / GO0001）+10^-99）+ -ln (P（CD0004 / GO0001）+10^-99)

Complex-score method:

Weighting the C-score, we will have value-positive (VP) and value-negative (VN)

For example,

	With GO0001	Without Go0001
CD0003	VP=C-score(CD0003/SG003)+ C-score(CD0003/SG004)	VN=C-score(CD0003/SG002)

Complex-score = ln (VP*VP/ (VN+0.5) +1)

To predict the unknown relationship between SG0005 and GO, we can calculate the Final-score by known relationship between SG0005 and CD.

Final-score =

Complex -score（CD0003/GO0001）+ Complex -score（CD0004/GO0001）

Naïve Bayes Classifier methods

Since we have got the domains in genes and the GO terms in genes, we can get the prior probability of having a single GO in a gene, and calculating for the posterior probability for $P(G_i|D)$, which stands for under the circumstances we've got domain D in a gene, the probability of having GO term G_i. Since we can have a list of GO terms each with a posterior probability, according to the Naïve Bayes classifier, we choose the one with max probability for such domain.

$$P(G_i|D)=\frac{P(D|Gi)*P(Gi)}{\sum_1^m P(D|Gi)*P(Gi)} \quad (1)$$

Data's preparation

1. Separate the know part into 90% and 10%, the 90% of known GO over genes can be used to be training samples, the left are to be tested for the power.

2. Establish a matrix for domains and GO terms, in which presenting the counts of over all the genes, how many times each domain has each GO term. The example of the matrix is presented below.

	G1	G2	…	Gm
D1				
D2				
…				
Dn				

3. The assumption of the model is based on that the domains and GO terms are independent variants. We can simplify the function 1 by the matrix. Summing up each column and divided by the number of unrepeated domains, we can get the value of prior probability of each GO term. $P(DG_i)$ is easy to obtain since each lattice stand for that value. With some basic transformation, we can get the second function

$$P(G_i|D)= \frac{P(DGi)}{\sum P(DGi)} \quad (2)$$

4. Calculate all the domains and pick up the GO term with max probability, then we can set a relation between a domain with a GO term which most likely to occur.

5. With a given new gene, with the domains listed, we can find a series of GO terms according to the step 4 and fulfill our prediction.

RESULT and DISCUSSION

ROC curves

In the figure2, there are 4 methods' ROC curves, besides fisher's test method, the rest ROC curves are pretty similar. AUC of the fisher's text method is the smallest, thus this method is not better than the other methods in this project.

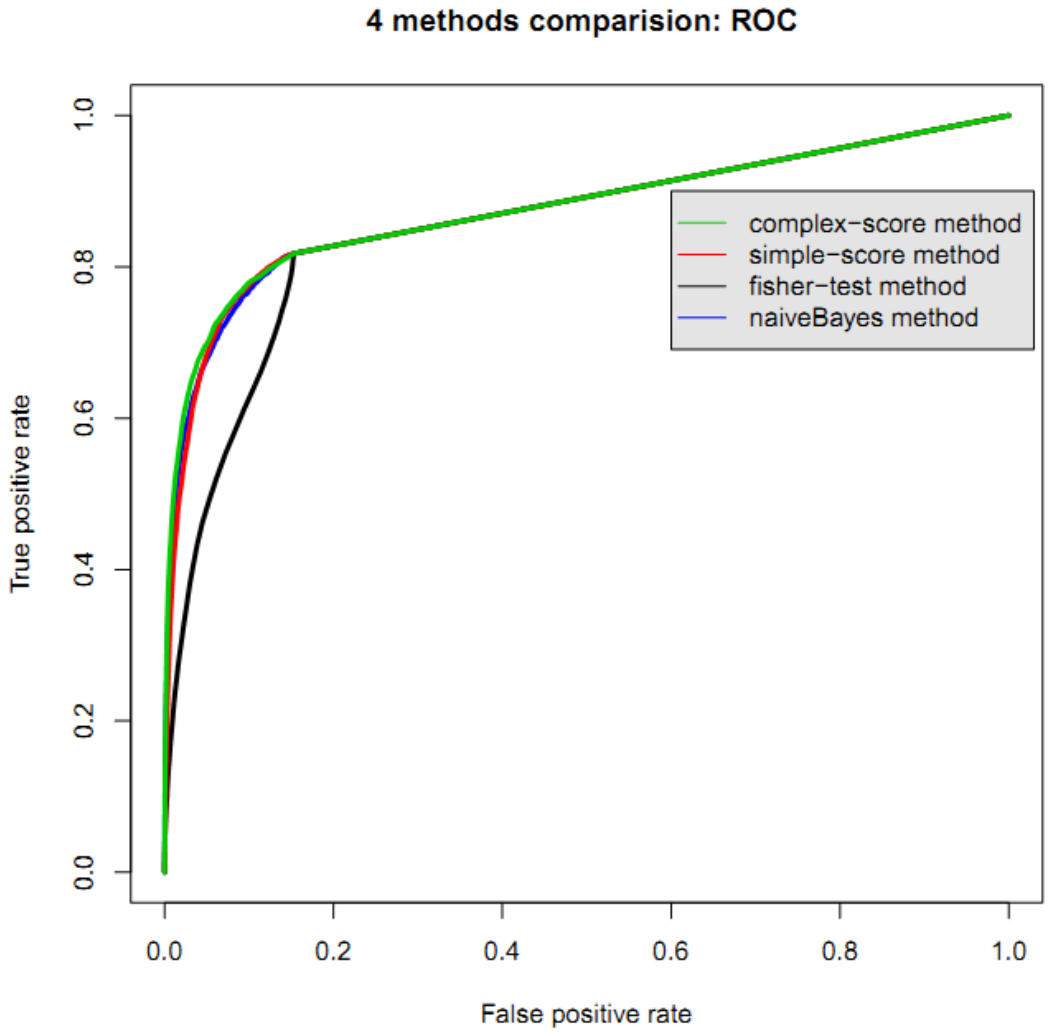


Figure2

Precision-recall curves

In the figure 3, there are 4 methods' Precision-recall curves, and we can see complex-score method is the best model in this project since it has the largest AUC.

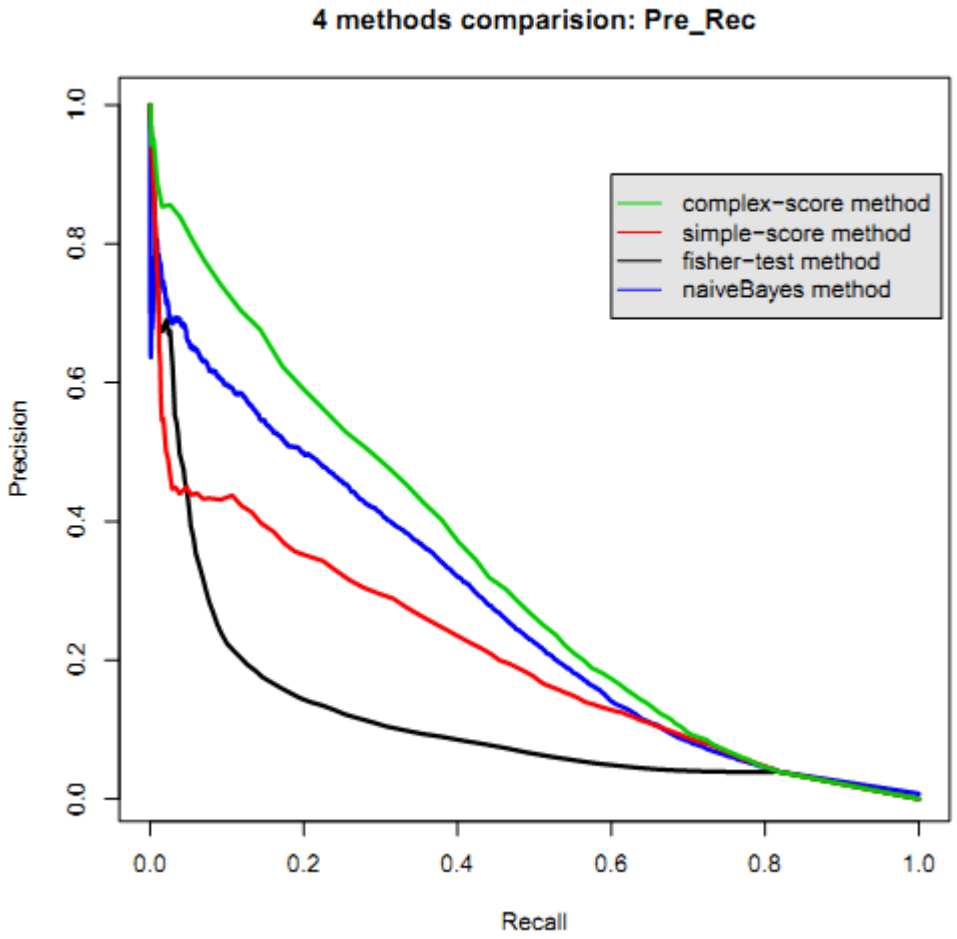
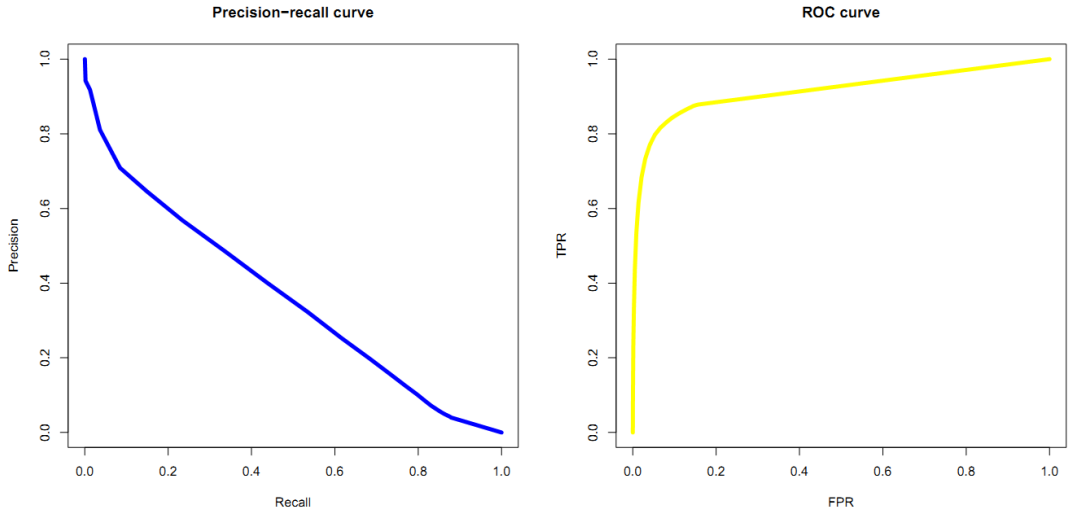


Figure3

20-fold validation

Then, we use 20-fold validation to assess the results of Complex-score method. We get the overall ROC curve and Precision-recall curves.



Final results

Finally, we use the Complex-score method to do the prediction of test genes by assigning probabilities (0-1) to sequences on whether it is likely to be assigned with the GO term. And the final result is in

/serverDNA/bachelor/LXW0147/Teamwork/final_result.txt

README FILE

All the scripts are in the fold: /serverDNA/bachelor/LXW0147/Teamwork.

#The fianl results is “final_result.txt”

#The scripts are in the corresponding folds

#The commands to run scripts

#Fisher_test_method

perl Prepare.pl &

perl GET_result.pl >result_fisher &

#Simple_score_method

perl Prepare.pl &

perl GET_result.pl >result_simple &

#Complex_score_method

perl Prepare.pl &

perl GET_result.pl >result_complex &

#NaiveBaye_method

Rscript Prepare.r &

Rscript GET_result.r &

#20-fold validation and make result matrix:

perl 20_fold_validation.pl >result &

perl make_result_matrix.pl >final_result.txt &

#plot

Rscript pre_rec.r &

Rscript roc.r &

Rscript plot.r &

Reference

[1] Marchler-Bauer A, Lu S, Anderson J B, et al. CDD: a Conserved Domain Database for the functional annotation of proteins[J]. Nucleic acids research, 2011, 39(suppl 1): D225-D229.

[2] Upton G J G. Fisher's exact test[J]. Journal of the Royal Statistical Society. Series A. Statistics in society, 1992, 155(3): 395-402.

[3] Ashburner M, Ball C A, Blake J A, et al. Gene Ontology: tool for the unification of biology[J]. Nature genetics, 2000, 25(1): 25-29.

[4] GENE ONTOLOGY CONSORTIUM, et al. The Gene Ontology (GO) database and informatics resource. *Nucleic acids research*, 2004, 32.suppl 1: D258-D261.

[5] Wu H, Su Z, Mao F, et al. Prediction of functional modules based on comparative genome analysis and Gene Ontology application[J]. Nucleic acids research, 2005, 33(9): 2822-2837.

[6] Forslund K, Sonnhammer E L L. Predicting protein function from domain content[J]. Bioinformatics, 2008, 24(15): 1681-1687.