12-1-2012

Data Mining of Pancreatic Cancer Protein Databases

Peter Revesz
University of Nebraska-Lincoln, prevesz1@unl.edu

Christopher Assi
University of Nebraska-Lincoln

Data Mining of Pancreatic Cancer Protein Databases

PETER Z. REVESZ
University of Nebraska-Lincoln
Computer Science and Engineering
358 Avery Hall, Lincoln, NE 68588
USA
cassi@cse.unl.edu

CHRISTOPHER ASSI
University of Nebraska-Lincoln
Computer Science and Engineering
256 Avery Hall, Lincoln, NE 68588
USA
revesz@cse.unl.edu

Abstract: Data mining of protein databases poses special challenges because many protein databases are non-relational whereas most data mining and machine learning algorithms assume the input data to be a type of relational database that is also representable as an ARFF file. We developed a method to restructure protein databases so that they become amenable for various data mining and machine learning tools. Our restructuring method enabled us to apply both decision tree and support vector machine classifiers to a pancreatic protein database. The SVM classifier that used both GO term and PFAM families to characterize proteins gave us over 73% accuracy in predicting whether a protein is involved in pancreatic cancer.

Key–Words: Pancreatic cancer, proteins, GO terms, PFAM families, data mining, decision trees, support vector machines

1 Introduction

Data mining is increasingly applied to non-relational databases [10, 12, 14, 19, 20]. The long-term goal of our research group is to develop data mining methods that are generally applicable to protein structure and function [11], protein evolution [18] as well as medical data [15, 16]. In the present paper, a preliminary version of which was presented in [1], we focus on a pancreatic cancer-related protein database, which was collected by Robert Powers and Bradley Worley, in the Department of Chemistry at the University of Nebraska-Lincoln, based on earlier pancreatic cancer research [3, 4, 5, 6, 9, 17, 22]. Pancreatic cancer was chosen as a test case because it has the lowest survival rate among different types of cancer. Data mining was used to investigate the relationship among anomalous proteins, which have unusually high or low levels in pancreatic patients. Early recognition of some patterns developing among these anomalous proteins may allow treatment to start earlier and increase the survival rate of pancreatic cancer patients.

Data mining of protein databases poses special challenges because many protein databases often contain set data types, whereas most data mining and machine learning algorithms assume relational database inputs. We overcame this problem by describing effecting ways to restructure the protein databases into relational databases. The restructured databases allowed the use of several types of classifiers, such as, Support Vector Machines (SVMs) and decision trees. Other types of data mining algorithms could be also used, but we chose these two types because they are currently the most frequently used data mining methods.

This paper is organized as follows. Section 2 describes some basic background. Section 3 presents the restructuring method and illustrates it on sample protein databases. Section 4 gives the results of applying the J48 decision tree and the libSVM classifiers to the restructured pancreatic cancer database. Finally, Section 5 gives our conclusions and possible directions for future work.

2 Background Concepts and Tools

Section 2.1 gives an introduction to classifiers and Section 2.2 describes the WEKA system that contains a library of implemented classifiers.

2.1 Classifiers

Let $R(x_1, \ldots, x_n, y)$ be a relation, where the set of attributes $X = \{x_1, \ldots, x_n\}$ is called the *feature space* and the y attribute is called a *label*. Each tuple of the relation describes some entity based on specific values of the feature space and the label. For example, each row may describe a protein with specific feature attributes, such as, molecular weight, amino acid sequence etc., and a label attribute, such as, whether it is involved in pancreatic cancer.

Given such a relation R, a classifier is mapping from X to y. If a classifier is correct on all tuples of relation R, then the value of y can be always predicted from the values of X. In practice, the classifier may not be correct on all proteins. Further, classifiers are intended to be able to classify even those proteins that are new, not just those that are already in R. Popular classifiers include *decision trees* and *Support Vector Machines* (SVMs). A decision tree is a tree which is read from the root towards the leaves, and whose internal nodes are tests and whose leaf nodes are categories [21]. For example, C4.5 is a well-known decision tree algorithm [13]. SVMs perform classification by constructing for relation R an n-dimensional hyperplane that optimally separates the data into two categories (for example when y=0 and y=1). An example of SVM is the libSVM implementation [8].

2.2 The WEKA Library

In our experiments we used the Waikato Environment for Knowledge Analysis (WEKA) system developed at the University of Waikato [2, 7]. WEKA provides an extensive library of data mining and machine learning algorithms. In WEKA, the input data is a relation or table which is represented by an Attributes Relation File Format (ARFF) file. Each ARFF file starts with a title to let the user know what kind of data is stored in the file. The title is followed by a relation type and then all the attributes and their types. Finally, the attribute declarations are followed by the actual data rows.

3 The Restructuring Method

In the pancreatic protein database collection of about eighty tables, we chose for our study the GO_np and PFAM_np tables, which contain data about pancreatic proteins that are not involved in cancer, and the GO_pdac and PFAM_pdac tables, which contains data about pancreatic proteins that are related to pancreatic cancer. GO_np had $70,331$, PFAM_np had $7,054$, GO_pdac had $30,888$, and PFAM_pdac had $7,272$ rows, that is, a total number of 125,545 rows. A simplified version of the GO_pdac looks as follows:

UID	GO
O43491	GO:0003779
O43491	GO:0005198
O43491	GO:0005886
O43491	GO:0008091
O43491	GO:0019898
O43491	GO:0030866
Q96C24	GO:0005215
Q96C24	GO:0005886
Q96C24	GO:0019898
Q96C24	GO:0030658
Q96C24	GO:0042043
⋮	⋮

Table 1: The GO_pdac table.

The GO_pdac table lists all (UID, GO) pairs, such that UID is the universal identifier of a pancreatic protein and GO is a feature descriptor, also called a GO term. There is a many-to-many relationship between the UIDs and the GO terms. For example, rows three and five with the same UID O43491 are related to two different GO terms, GO:0005886 and GO:0019898. On the other hand, rows three and eight with the same GO term GO:0005886 are related to two different UIDs, O43491 and Q96C24.

The GO_np tables listed (UID, GO) pairs of non-pancreatic proteins. We merged the GO_np and GO_pdac tables without losing the information whether the protein is related to cancer or not. Hence we extended the GO_np and the GO_pdac tables with a Y column, which denotes whether the protein is related to pancreatic cancer or not. All the proteins in the GO_np table are extended with a Y value of ”0”, while all the proteins in the GO_pdac table are extended with a Y value of ”1” as follows:

```
create view GO_merge (UID, GO, Y) as
select UID, GO, 0 from GO_np
union
select UID, GO, 1 from GO_pdac;
```

After the above query is executed the GO_merge table looks as follows:

UID	GO	Y
O43491	GO:0003779	1
O43491	GO:0005198	1
O43491	GO:0005886	1
O43491	GO:0008091	1
O43491	GO:0019898	1
O43491	GO:0030866	1
Q96C24	GO:0005215	1
Q96C24	GO:0005886	1
Q96C24	GO:0019898	1
⋮	⋮	⋮

Table 2: The GO_merge table.

We restructured or *“flattened”* the above table by an SQL query that transformed GO_merge into another table GO_merge_flat in which all information about a single protein appears in one row, as shown in Table 3.

UID	3779	5198	5215	5886	8091	19898	30866	Y
O43491	1	1	0	1	1	1	1	1
Q96C24	0	0	1	1	0	1	0	1

Table 3: The GO_merge_flat table.

In theory, the number of attributes in the restructured relation is n+2, where n is the number of distinct GO terms. Apart from UID and Y, these distinct GO terms form the attributes of the restructured relation. Below each GO term a 1 or 0 indicates whether the GO term applies to the protein indicated by the UID on the left.

In practice, we cannot actually restructure the entire GO_merge table because there are 7935 GO terms. Moreover, most of these GO terms occur very infrequently. Hence we selected only the top 200 most frequent GO terms as follows. First we found the frequency of each Go terms using the following SQL query:

```
create view GOcount(GO,count) as
select GO, count(*)
from GO_merge
group by GO;
```

The new table GOcount(GO,count) contains the count of each GO term. We extracted the top 200 most frequent GO terms into a text file as follows:

```
select GO from GOcount
order by count desc limit 200
into outfile '/tmp/MergeTop200GO.txt';
```

We wrote a C++ program to automatically generate the restructuring SQL query. Apart from some initialization and ending, the program repeatedly reads the next GO term from the input file MergeTop200GO.txt and writes to an output file SQL_flatten.txt the line of the SQL query that corresponds to the GO term. Below is how the SQL_flatten.txt file looks like.

```
select UID,
max(case when GO = 'GO:0016021' then 1 else 0
end) as 'GO:0016021',
max(case when GO = 'GO:0005515' then 1 else 0
end) as 'GO:0005515',
max(case when GO = 'GO:0005634' then 1 else 0
end) as 'GO:0005634',
max(case when GO = 'GO:0005737' then 1 else 0
end) as 'GO:0005737',
max(case when GO = 'GO:0008270' then 1 else 0
end) as 'GO:0008270',
max(case when GO = 'GO:0006350' then 1 else 0
end) as 'GO:0006350',
max(case when GO = 'GO:0007165' then 1 else 0
end) as 'GO:0007165',
max(case when GO = 'GO:0005886' then 1 else 0
end) as 'GO:0005886',
max(case when GO = 'GO:0005524' then 1 else 0
end) as 'GO:0005524',
max(case when GO = 'GO:0003677' then 1 else 0
end) as 'GO:0003677',
⋮
Y
from GO_merge
group by UID
```

When the SQL query is executed, for each UID it checks all the GO terms. If any of the GO terms the UID is associated with matches a particular GO term for which we are creating a column in the flattened table, then that GO term will get a value of "1" else it will get a value of "0". The process then continues until it does not read any more UID groups.

3.1 Merging GO_merge and PFAM_merge

The PFAM table is similar to the GO table. The PFAM table contains the UID of proteins and the PFAM terms, which form another set of characterizations of proteins as an alternative to the GO term characterization. We can create PFAM_merge by merging PFAM_np and PFAM_pdac similarly to how we created GO_merge. Figure 1 outlines the process of merging the GO_merge and the PFAM_merge tables together when we need to use both the GO and the PFAM terms.

Below is an example PFAM_merge table.

UID	**family**	**Y**
P02656	PF05778	0
P09651	PF00076	0
Q9BY79	PF00431	0
Q9BY79	PF01392	0
Q9BY79	PF00057	0
O95931	PF00385	0
Q9UKU0	PF00501	0
P10323	PF00089	0
Q17RR3	PF00151	0
Q17RR3	PF01477	0
⋮	⋮	⋮

Table 4: The PFAM_merge table.

In Figure 1, SQL 3 refers to the following query:

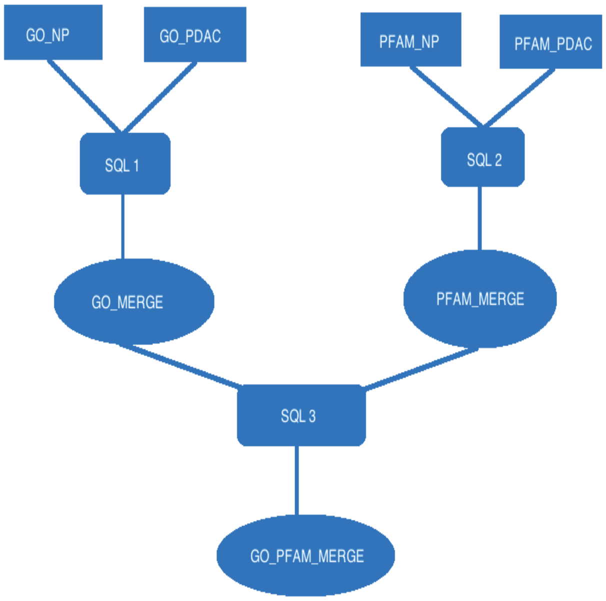

Figure 1: GO_PFAM_Merge

```
SELECT T.UID,
max(case when GO = 'GO:0016021' then 1 else 0
end) as 'GO:0016021',
⋮
max(case when family = 'PF07647' then 1 else 0
end) as 'PF07647'
⋮
, T.Y
FROM GO_merge T JOIN PFAM_merge ON T.UID
= PFAM_merge.UID
group by UID
```

In our experiments, we used the top n most frequent GO terms as well as the top m most frequent PFAM terms, yielding a relation with $n + m + 2$ attributes. We varied the values of n and m as described in the next section.

4 Experimental Results

Given a flattened file, as in Table 3, it is easy to generate an ARFF file, which is needed for the WEKA system. In the ARFF file, the UID attribute ranges over strings that describe protein IDs, and the ”relation” attribute substitutes for the ”Y” attribute. For example, Table 3 is described using ARFF as follows:

```
@relation GO_merge_flat
@attribute “UID” {O43491, Q96C24}
@attribute “GO:0003779 { 0, 1}
@attribute “GO:0005198 { 0, 1}
@attribute “GO:0005215 { 0, 1}
@attribute “GO:0005886 { 0, 1}
@attribute “GO:0008091 { 0, 1}
@attribute “GO:0019898 { 0, 1}
@attribute “GO:0030866 { 0, 1}
@attribute “relation” { 0, 1}
@data
“O43491”,1,1,0,1,1,1,1,1
“Q96C24”,0,0,1,1,0,1,0,1
```

From our WEKA library, we used both libSVM support vector machines, which was previously added to the library, and J48 decision trees. Both of these accepted input in ARFF format. The stratified cross-validation was used in all our classifications.

libSVM Support Vector Machine: Using libSVM with the GO_merge_flat file, WEKA gave the following:

Correctly Classified Instances	12947	72.1563%
Incorrectly Classified Instances	4996	27.8437%
Total Number of Instances	17943	

WEKA also gave the following confusion matrix:

a	b	classified
12794	305	$a = 0$
4691	153	$b = 1$

The confusion matrix displays the relationship between two or more categorical variables. The number of correctly classified instances is the sum of the diagonals in the confusion matrix; all the others are incorrectly classified. For libSVM with the PFAM_merge file and stratified cross-validation, the data mining results with were as follows:

Correctly Classified Instances	11590	71.707%
Incorrectly Classified Instances	4573	28.293%
Total Number of Instances	16163	

The classification for all our instance was for about 71.7% of the instances. Below is the confusion matrix:

a	b	classified
163	4263	$a = 0$
310	11427	$b = 1$

J48 Decision Tree: Our next set of experiments used the J48 decision tree. The decision tree with the GO_merge_flat file gave the following results:

Correctly Classified Instances	12922	72.0169%
Incorrectly Classified Instances	5021	27.9831%
Total Number of Instances	17943	

The classification was again about 72% correct. Below is the confusion matrix for the J48 decision tree:

a	b	classified
12562	537	$a = 0$
4484	360	$b = 1$

For decision tree with the PFAM_merge_flat file, the data mining results were as follows:

Correctly Classified Instances	11719	72.5051%
Incorrectly Classified Instances	4444	27.4949%
Total Number of Instances	16163	

The classification for all our instance was correct over 72%. It was slightly better than for GO_merge_flat with the decision tree classification. Below is the confusion matrix for the PFAM_merge decision tree:

a	b	classified
144	4282	$a = 0$
162	11575	$b = 1$

4.1 Improving the Accuracy

As we saw above, for both the GO_merge_flat and the PFAM_merge_flat files and both the libSVM and the J48 the accuracy was around 72%. A natural question is whether the accuracy can be improved by using both the GO terms and the PFAM families together. As we saw in Figure 1, these terms can be combined in a relation GO_PFAM_merge. This file can be also flattened and represented in ARFF. We performed another set of experiments using WEKA and the GO_PFAM_merge_flat file. The results for libSVM were the following:

Correctly Classified Instances	13099	73.0034%
Incorrectly Classified Instances	4844	26.9966%
Total Number of Instances	17943	

Finally, the results for J48 were the following:

Correctly Classified Instances	12936	72.095%
Incorrectly Classified Instances	5007	27.905%
Total Number of Instances	17943	

Our results from the GO_PFAM_merge analysis show that the libSVM has the highest percentage of 73% compare to 72% for the decision tree.

4.2 Discussion of the Results

The results reveal that the characterizations of the pancreatic proteins by either GO terms or PFAM families can be used to predict with a good, that is, around 72%, accuracy whether they are involved in cancer. Since the characterizations of proteins is mainly based on their biological functions, the results imply that the likelihood of a protein being involved in cancer depends on its particular functions. Although the 72% accuracy is interesting, for medical applications a higher, over 90%, accuracy would be necessary. It is not clear how that higher accuracy could be achieved. Our second set of experiments with both GO terms and PFAM families together gave a slight increase in accuracy to 73% in the case of libSVM. It is possible that by adding even more protein attributes, the accuracy of classification would improve further.

5 Conclusions and Further Work

The result that the functional characterizations of proteins by either GO terms or PFAM families enable a good prediction of pancreatic cancer link may be also generalized to other types of cancers. It appears that proteins involved in certain functions within cells are more likely to be associated with cancer. Biologists could investigate further the cancer-related functions and may improve the results to develop an early detection method for pancreatic cancer enabling earlier treatment and thereby increase the survival rate of pancreatic patients.

Note: Since graduation from the University of Nebraska-Lincoln, Christopher Assi found employment with the U.S. federal government. Peter Revesz was awarded an AAAS Science & Technology Policy Fellowship and as part of the fellowship program took a leave of absence from the University of Nebraska-Lincoln to serve as a grants Program Manager in the U.S. Air Force Office of Scientific Research (AFOSR). The views and opinions expressed in this publication are those of the authors and do not necessarily reflect the official policy or position of any agency of the U.S. government.

References:

[1] C. Assi, *Data Mining of Protein Databases*, M.S. Thesis, University of Nebraska-Lincoln, August 2012.

[2] R. Bouckaert, E. Frank, M. Hall et al., *WEKA Manual*, The University of Waikato, Version 3-7-3, 2010.

[3] A. Brazma, H. Parkinson, U. Sarkans et al., ArrayExpress - A public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 31, 2003, pp. 68-71.

[4] R. Chen, E. C. Yi, S. Donohoe, S. Pan et al., Pancreatic cancer proteome: The proteins that underlie invasion, metastasis, and immunologic escape. *Gastroenterology*, 129, 2005, pp. 1187-97.

[5] T. Crnogorac-Jurcevic, R. Gangeswaran, V. Bhakta and G. Capurso, Proteomic analysis of chronic pancreatitis and pancreatic adenocarcinoma. *Gastroenterology*, 129, 2005, pp. 1454-63.

[6] R. Grutzmann, C. Pilarsky, O. Ammerpohl et al., Gene expression profiling of microdissected pancreatic ductal carcinomas using high-density DNA microarrays. *Neoplasia*, 6, 2004, pp. 611-22.

[7] M. Hall, E. Frank, G. Holmes, et al., The WEKA Data Mining Software: An Update.

[8] C. Hsu, C. Chang and C. Lin, A Practical Guide to Support Vector Classification. National Taiwan University, Taipei 106, Taiwan. 2003.

[9] S. Jones, X. Zhang, D. W. Parsons, J. C. Lin et al., Core signaling pathways in human pancreatic cancers revealed by global genomic analyses, *Science*, 321, 2008, pp. 1801-6.

[10] P. C. Kanellakis, G. Kuper, P. Z. Revesz, Constraint Query Languages, *Journal of Computer and System Sciences*, 51(1), 1995, pp. 26-52.

[11] R. Powers, J. Copeland, K. Germer, K. Mercier, V. Ramanathan and P. Z. Revesz, Comparison of Protein Active-Site Structures for Functional Annotation of Proteins and Drug Design, *Proteins: Structure, Function, and Bioinformatics*, 65(1), 2006, pp. 124-135.

[12] T. S. K. Prasad et al., Human Protein Reference Database 2009 Update. *Nucleic Acids Research*. 37, 2009, D767-72.

[13] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993.

[14] P. Z. Revesz, *Introduction to Databases: From Biological to Spatio-Temporal*, Springer-Verlag, New York, 2010.

[15] P. Z. Revesz and T. Triplet, Classification Integration and Reclassification using Constraint Databases, *Artificial Intelligence in Medicine*, 49(2), 2010, pp. 79-91.

[16] P. Z. Revesz and T. Triplet, Temporal Data Classification using Linear Classifiers, *Information Systems*, 36(1), 2011, pp. 30-41.

[17] J. Shen, M. D. Person, J. Zhu, J. L. Abbruzzese, D. Li, Protein expression profiles in pancreatic adenocarcinoma compared with normal pancreatic tissue and tissue affected by pancreatitis as detected by two-dimensional gel electrophoresis and mass spectrometry. *Cancer Research*, 64, 2004, pp. 9018-26.

[18] M. Shortridge, T. Triplet, P. Z. Revesz, M. Griep, R. Powers, Bacterial Protein Structures Reveal Phylum Dependent Divergence, *Computational Biology and Chemistry*, 35(1), 2011, pp. 24-33.

[19] B. Thuraisingham, A Primer for Understanding and Applying Data Mining, *IT Professional*, 2000, pp. 28-31.

[20] T. Triplet, M. Shortridge, M. Griep, J. Stark, R. Powers, P. Z. Revesz, PROFESS: a PROtein Function, Evolution, Structure and Sequence database, *Database – The Journal of Biological Databases and Curation*, doi no. 10.1093/baq011, 2010.

[21] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[22] M. Yamada, K. Fujii, K. Koyama, S. Hirohashi and T. Kondo, The Proteomic Profile of Pancreatic Cancer Cell Lines Corresponding to Carcinogenesis and Metastasis. *Journal of Proteomics and Bioinformatics*, 2, 2009, pp. 18.