ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/ijmi

Gene functional annotation by statistical analysis of biomedical articles

T. Theodosiou[a,*], *L. Angelis*[a], *A. Vakali*[a], *G.N. Thomopoulos*[b]

[a] *Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece*
[b] *Department of Biology, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece*

ARTICLE INFO

Article history:
Received 20 August 2005
Received in revised form
8 March 2006
Accepted 26 April 2006

Keywords:
Genes annotation
Gene ontology
Linear discriminant analysis
Classification
Text mining

ABSTRACT

Background: Functional annotation of genes is an important task in biology since it facilitates the characterization of genes relationships and the understanding of biochemical pathways. The various gene functions can be described by standardized and structured vocabularies, called bio-ontologies. The assignment of bio-ontolgy terms to genes is carried out by means of applying certain methods to datasets extracted from biomedical articles. These methods originate from data mining and machine learning and include maximum entropy or support vector machines (SVM).

Purpose: The aim of this paper is to propose an alternative to the existing methods for functionally annotating genes. The methodology involves building of classification models, validation and graphical representations of the results and reduction of the dimensions of the dataset.

Methods: Classification models are constructed by Linear discriminant analysis (LDA). The validation of the models is based on statistical analysis and interpretation of the results involving techniques like hold-out samples, test datasets and metrics like confusion matrix, accuracy, recall, precision and *F*-measure. Graphical representations, such as boxplots, Andrew's curves and scatterplots of the variables resulting from the classification models are also used for validating and interpreting the results.

Results: The proposed methodology was applied to a dataset extracted from biomedical articles for 12 Gene Ontology terms. The validation of the LDA models and the comparison with the SVM show that LDA (mean *F*-measure 75.4%) outperforms the SVM (mean *F*-measure 68.7%) for the specific data.

Conclusion: The application of certain statistical methods can be beneficial for functional gene annotation from biomedical articles. Apart from the good performance the results can be interpreted and give insight of the bio-text data structure.

1. Introduction

The functional annotation of genes is an important issue in biology, needed for understanding and characterizing the cells and organisms (at the molecular level). Understanding gene's function is crucial since the genes are responsible, through the synthesis of RNA and proteins, for the proper function of a cell [1,2].

Biologists have defined the so-called bio-ontologies to provide a standard, controlled and structured vocabulary for describing a gene function. Bio-ontologies are sets of defined and networked biological terms/words. Several bio-

* *Corresponding author*. Tel.: +30 6946454141.
E-mail address: theodos@csd.auth.gr (T. Theodosiou).
1386-5056/$ – see front matter © 2006 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.ijmedinf.2006.04.011

logical ontologies, describing gene function exist, such as the EcoCyc [3] and the most popular Gene Ontology (GO) [1,2,4–6]. Under a bio-ontology, the terms and their relationships are precisely defined by using certain distinct "codes" to describe biological phenomena with a common language, which in turn can easily be processed and implemented by nowadays computer systems.

The main problem in manually assigning a GO code to a gene is the enormous amount of available information, since the functional information of a gene product and its relevant gene can be found inside many various sources (published literature, sequence analysis results, 3D analysis of the products of the gene, etc.). A conventional approach is to have a human expert searching through myriads of publicly available published articles (in sources such as the PubMed [7]) and validating the information inside articles "against" the actual experimentation data. It should be noted that it is very difficult to discover and analyse all the relationships of genes functions from articles sources, due to the following facts:

- Lack of information: sometimes the function of a gene is either not clear or not adequately described. In this context, the GO Consortium[1] has improvised the evidence codes [1,2], that show the quality level of an annotation of a gene to a particular term.
- GO codes variation: the GO codes may change, expand and undergo refinement since the biomolecular related research and understanding is quite emerging and evolving. As a result the reassignment of GO codes to genes [8] might be required.
- New genes discovery: the number of genes is rapidly increasing, especially now that the DNA sequence of whole genomes for a variety of organisms is freely available [9].

From the above, it is obvious that the manual assignment of a GO code to known genes is a labour-intensive, and time-consuming task [8,10] and already several computational approaches have been proposed (such as GO figure [11] and GOtcha [12]). These computational methods have been shown to be prone to errors and they are not considered as accurate and reliable as the human annotation of genes (performed by specialized biologists) [2]. Therefore, it is essential to improve the effectiveness of applied computational methods by increasing their reliability and performance.

1.1. *Related work*

Several earlier research efforts have focused on annotating genes by Natural Language Techniques on published biomedical literature [10], where instead of GO codes, they use informative keywords [13,14] extracted from the literature or predefined codes/terms [15,16]. The main motive behind these attempts was the mining, retrieval and utilization of the vast knowledge existing in the literature for the benefit of automatic gene annotation.

GO codes usage has been quite popular in the literature [6,8,17–19]. Such research efforts have focused on the following issues:

- Using sequence similarity, between genes or proteins, in order to assign a GO code to a gene or protein [6,17,18].
- Application of data mining and machine learning methods that can facilitate the process of assigning a GO code to a gene, using the information inside published biomedical literature [8,19]. In [8], three different classification models were compared on biomedical articles corresponding to 21 GO codes for the assignment of a function to a gene. The compared models were maximum entropy, naive Bayes and nearest-neighbourhood classifiers. The authors concluded that maximum entropy, having an average performance of 72%, outperformed the other two methods. In [19], support vector machines (SVM) were applied to articles from 12 GO codes and yeast GO Slim terms[2] from SGD [20] in order to develop a classification model. The authors concluded that SVM outperformed maximum entropy classification since the *F*-measure they used for assessing the performance was 67% for SVM and 49% for the maximum entropy classifier.

1.2. *Motivation and contribution*

The motivation for the present work was to examine whether a classical multivariate statistical method such as LDA, which has been used successfully in a wide range of applications, can perform well in this specific problem. LDA is a well-known method with strong statistical hypothesis-testing background for discrimination and classification and has a lot of interesting properties and outcomes. Its basic feature is the simplicity in the description of the difference of two or more groups based on linear models built from a sample. It was found to perform very well in comparison with methods like SVM and other 33 classification algorithms, new and old, as presented in [21]. In general, LDA is competitive with newer and more sophisticated methods, like SVM [22].

Furthermore, the developed models may serve not only as means for classification of new cases but also for the generation of new variables which can provide better insight of our data. Indeed, a typical problem with the data extracted from texts and represented by numerical vectors, is the large numbers of variables (usually thousands), which is prohibiting for any visual inspection of the data. On the other hand, the visual representation is very important not only for highlighting interesting patterns in the data, but also for validating a procedure's results (see [23] for a useful discussion from the user's perspective). More specifically, in terms discrimination and classification problems it is useful to represent the different categories by different patterns in order to classify new cases visually by their scheme.

Based on the aforementioned issues, we believe that the application of LDA to the specific problem of gene annotation is appealing not only from the researcher's point but from the user's as well.

The main contribution of this work is the presentation of a general statistical framework suitable for gene annotation

[1] http://www.geneontology.org.

[2] A subset of GO categories which gives a broader view of gene functions [1,2].

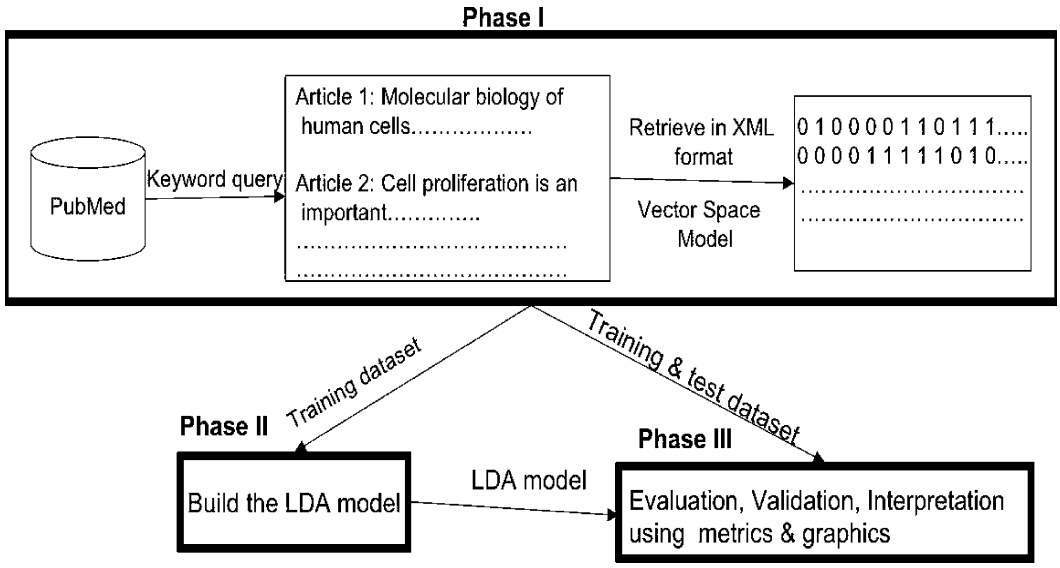

Fig. 1 – The three phases of the proposed methodology.

using published articles, which can be summarized in the following points:

- Use of a multivariate statistical methodology based on LDA for discriminating large corpuses of documents corresponding to different GO codes and for classification of new documents.
- Use of performance metrics and graphical tools for validation of the classification results of LDA.
- Graphical representation of the GO codes based on the variables created by LDA.

The proposed framework is illustrated by an application to real data. The data are also used for comparing the classification accuracy between LDA and SVM.

1.3. *Article outline*

The remainder of the paper is structured as follows: Section 2 describes the methodology used, whereas Section 3 presents the datasets used and reports the results. Finally, in Section 4 there is a discussion about the presented methodology and directions for future work.

2. Methodology

The application of the statistical methodology to articles for gene annotation is part of a more general procedure, which involves the initial retrieval of a corpus of documents and their transformation to numerical vectors manageable by statistical software. The entire procedure consists of three main phases shown in Fig. 1 and described briefly below:

- Dataset creation (Phase I): retrieval of relevant to GO codes documents, processing and conversion into computer readable format. Construction of the training and the test dataset.
- Building the statistical model (Phase II): building the classification model by LDA using the training dataset.
- Analysis of the results (Phase III): validation and interpretation of the classification results using mainly the test dataset.

2.1. *Document retrieval, processing and conversion*

The retrieval of the proper biological documents, their processing and their conversion (identified as Phase I) involves extracting published biomedical abstracts (title and other publication details) from the popular PubMed database queried by certain keywords. A certain set of query keywords that can enhance the search and improve the accuracy are the Medical Subject Headings (MeSH) terms [24].

It must be noted that MeSH and GO terms serve a different purpose. The goal of MeSH terms is to index articles in the PubMed database, whereas the purpose of GO terms is to describe the functions of a gene product. Nevertheless, some MeSH terms could be used as a proxy for GO terms (Table 1) [8]. In cases where the availability of MeSH terms is questionable, there are techniques as in [25] that can relate the two types of terms.

Upon retrieval, the next step is the transformation of documents[3] in a format suitable to reduce their complexity, and represent them for use in computational tasks. The most common document transformation is based on Vector Space Model (VSM) [26,27]. According to VSM, a document is represented as a vector of specific weighted words. The process involves the following steps [28,29]:

1. Extract all words from the entire set of documents (ignoring case). This process is called tokenization.

[3] In our context a document or an article refers to the title and the abstract of a biomedical article.

Table 1 – The 12 queries used to extract the abstracts relevant to each GO code for the training dataset

Biological term	GO code	Group no.	PubMed query
Autophagy	GO:0006914	1	(autophagy [TI] OR autophagocytosis [MAJR]) and (Proteins[MH] OR Genes[MH]) and 1940:1999[DP]
Cell cycle	GO:0007049	2	(cell cycle[MAJR]) and Genes[MH] and 1996:1999[DP]
Cell death	GO:0008219	11	(cell death[MAJR]) and Genes[MH] and 1997:1999[DP]
Cell proliferation	GO:0008283	3	(cell proliferation[TI]) and Genes[MH] and 1940:1999[DP]
Cell–cell signalling	GO:0007267	4	(synaptic transmission[MAJR] or synapses[MAJR] or gap junctions[MAJR]) and Genes[MH] and 1940:1999[DP]
Chemimechanical coupling	GO:0006943	5	(contractile proteins[MAJR]) and Genes[MH] and 1993:1999[DP]
Meiosis	GO:0007126	6	(meiosis[MAJR]) and (Genes[MH] or Proteins[MH]) and 1986:1999[DP]
Metabolism	GO:0008152	7	(metabolism[MAJR]) and Genes[MH] and 1989:1999[DP]
Oncogenesis	GO:0007048	8	(cell transformation, neoplastic[MAJR]) and Genes[MH] and 1994:1999[DP]
Signal transduction	GO:0007165	12	(signal transduction[MAJR]) and Genes[MH] AND 1995:1999[DP]
Stress response	GO:0006950	9	(wounds[MAJR] or DNA repair[MAJR] or DNA damage[MAJR] or Heat-Shock response[MAJR] or stress [MAJR] or starvation[TI] or soxR[TI] or (oxidationreduction[MAJR] NOT Electron-Transport[MAJR])) and Genes[MH] AND 1996:1999[DP]
Transport	GO:0006810	10	(biological transport[MAJR] or transport[TI]) and Genes[MH] and 1985:1999[DP]

The test dataset queries differ only in the date of publication ([DP])

2. Eliminate non-content-bearing, non-informative words, called "stopwords" such as "a", "and", "the", etc. The stopwords the algorithm uses are the same as the ones used in PubMed[4].
3. Use only the root of each word. This process is called stemming. The algorithm we used for stemming is based on [30] and is implemented in Perl by Mary D. Taffet[5].
4. Count the number of occurrences of each word for each document.
5. Eliminate "high-frequency" (appearing in more than 95% of the total number of articles) and "low-frequency" (appearing in less than 0.05% of the total number of articles) words.
6. Construct a vector with weights for all of the remaining words.

The optimal weighting scheme depends on the relevant articles for each GO code [19], and it is not possible to know it a priori [31]. The simplest weighting scheme is the one assigning weight "1" for existing words ("0" otherwise) in the document (as in Eq. (1)). There also are weighting schemes [28,29,31], that use more complicated weight values, but here we used the simplest Boolean value weight to avoid cumbersome calculations, which do not guarantee better performance [31].

Therefore, after the processing, each document d is represented as a p-dimensional vector $(x_{d1}, x_{d2}, \ldots, x_{dp})$, where x_{di} is the weight of term t_i within the document d. For the binary weighting scheme each weight is:

$$x_{di} = \begin{cases} 1 & \text{if } t_i \in d \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

At the end of Phase I, the different document vectors are used for constructing the datasets. Note that due to step 5, it is possible to have some vectors with zero elements which are excluded from the datasets.

2.2. *Classification methods*

The classification methods used for our experimentation were LDA and SVM. SVM were used for comparison with LDA since it has been shown in [19] to perform very well on the problem of functionally annotating genes. Despite the fact that both in discriminant analysis and in SVM several non-linear approaches appear in literature ([32,33]), we have chosen the linear model for discriminant analysis and the linear kernel for SVM. This is due to their simplicity and ease of implementation, which is one of the goals of our methodology.

2.2.1. *Support vector machines (SVM)*

SVM is a typical binary classifier, which may only solve classifications problems of two groups. In case of (more than two) multi-groups classification the standard approach is to reduce the multi-class problem into several binary subproblems [34].

SVM method has already been used for similar classification problems as in our case (such as in [19]). The basic idea in SVM is to define an optimal separating hyperplane that could separate the observations into two classes. This hyperplane is constructed based on certain vectors of the dataset (the so-called support vectors) and a proper (so-called) kernel function (to project the original data to this hyperplane). Typically, this kernel function requires several parameters which should be appropriately tuned. Examples of such parameters are the degree of the polynomial (to be set in polynomial kernel), or the value of the gamma parameter (to be set in radial kernel). The simplest form of the SVM has a linear kernel, and it is used as a basis for comparisons with our proposed linear discriminant analysis approach.

2.2.2. *Linear discriminant analysis (LDA)*

Linear discriminant analysis introduced by Fisher [35,36], is used to model the relationship between a dependent cate-

[4] http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=helppubmed.table.pubmedhelp.T43.

[5] http://www.comp.lancs.ac.uk/computing/research/stemming/Files/Perl.zip.

gorical variable and a set of independent or exploratory variables. In our context the independent variables represent the existence of words whereas the categorical dependent variable is their corresponding GO code with k possible values categories.

The main goals of LDA are (a) to explain the differences between the classes (in our case the GO codes), in terms of the independent variables; and (b) to utilize the model for future predictions. The method along with the prediction model provides a number of statistical results leading to the estimation of probability (posterior probability) that a vector of word weights belongs in a particular group.

The idea is to find a linear combination of the independent variables that would produce maximally different discriminant scores z between groups (GO codes). If $\boldsymbol{a}$ denotes the linear combination and $\boldsymbol{X}$ an $n \times p$ data matrix, then the discriminant scores are given by

$$\mathbf{z} = \boldsymbol{X}\boldsymbol{a} \tag{2}$$

In our context the rows of the data matrix $\boldsymbol{X}$ contain the binary vector representation of the retrieved documents. According to (1) the matrix $\boldsymbol{X}$ can be written as

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots \\ x_{d1} & \cdots & x_{dp} \\ \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} \tag{3}$$

We also assume that the n row vectors of the $\boldsymbol{X}$ matrix fall in k categories (GO codes).

Fisher proposed to choose the linear combination $\boldsymbol{a}$ that maximizes the ratio of the between-group sum of squares to the within-group sum of squares of the discriminant scores z. This is an eigenvalue problem, which is solved numerically by using singular value decomposition (SVD) [36]. In the case where the number of groups is only two, then only one discriminant function is enough to distinguish the groups. If the number of groups k is greater than two ($k > 2$), then we use $m = \min(k-1, p)$ discriminant functions, called canonical discriminant functions, to exhibit the differences among the groups.

In order to classify new observations (documents relevant to specific genes) we estimate the probability of a new observation (weighted vector of words) to belong in a specific category (GO code). This is achieved using the Mahalanobis distance and the Bayes's theorem [36]. Having estimated the aforementioned probabilities separately for each GO code, we can finally decide to classify the new observation to the code with the largest probability.

Furthermore, the use of probabilities allows us to investigate the possibility that a gene may belong to another GO code other than the one with the highest probability. That is, if there is uncertainty regarding the correct classification of a gene, it may be helpful to consider as possible classification the GO code with the second, the third, etc. largest probability. This issue is further discussed in the experimentation section.

Another interesting point is that LDA itself produces weights for each word in the form of coefficients of the linear discrimination functions. These weights signify the importance of each word in the classification model. It is also noteworthy that LDA can be applied either to the original data matrix or to the data resulting from a standard data reduction technique, for example principal component analysis (PCA) or more generally, factor analysis [32,37,38].

2.3. *Validation methods*

2.3.1. *Graphical analysis of a dataset*

Graphical analysis is very important, because it can provide new insights to our data and reveal various groupings and correlations [39]. Since in our datasets, the dimensionality of data is very large there is a need for advanced multivariate graphical techniques in order to efficiently depict the resulted data properties. The following graphical methods have been used in the present work:

- Boxplots [38]: the empirical distribution of a variable is plotted separately for groups of cases. They are useful in depicting differences among groups with respect to a single variable.
- Scatterplots [38]: a relatively simple method to represent combinations of variables is to use a matrix of scatterplots, which can simultaneously plot several pairs of variables and visualize their relations in a two dimensional space. The scatterplots can reveal various groupings in the data and also serve as diagnostic tools for validating the performance of the classification methods.
- Andrews' curves [40]: each multivariate observation (x_1, x_2, x_3, ...) is transformed into a curve based on the following function:

$$f(t) = x_1 \frac{1}{\sqrt{2}} + x_2 \sin(t) + x_3 \cos(t) + x_4 \sin(2t) + x_5 \cos(2t) + \cdots$$

This function is plotted over the range $-\pi < t < \pi$ and has several beneficial characteristics, like it preserves the mean, the standard deviation, the distances, etc. These curves are representative of the dataset and can assist in distinguishing different groups, outliers, etc.

Since our datasets are very large there is an obvious problem of representing with clarity all the data cases in scatterplots and Andrews' curves. It is therefore necessary to simplify the pictorial representation by using the centroids of the different groups. The centroids of the separate groups are multidimensional points having coordinates the means for all the variables within the group [37,38].

2.3.2. *Performance metrics*

The validation of the model's sensitivity (changes in the accuracy) can be achieved by the repeating random drawing of sub-samples (hold-out samples) [36] from the training dataset. These samples split the initial set into a subset containing most of the observations used for training the model and another smaller for validating the classification performance.

The size of the hold-out samples is determined empirically (there is no standard formula) and usually is not larger than 30% of the original dataset [38].

The basis for the computation of various performance measures is the confusion or classification matrix [38]. Specifically, the most common performance measures are:

(1) Accuracy [28] of the classification is the ratio of the correct classified documents to the total number of documents in the dataset.
(2) Precision [28] is the percentage of the correct classifications of a specific group to all the documents assigned to that group.
(3) Recall [28] is the percentage of the correct classifications of a specific group to all the documents of the group contained in the dataset.
(4) F-measure [41] is the harmonic mean between precision and recall.

2.4. *Computation*

All the computations were performed on an Intel Pentium IV computer running at 2.8 GHz with 1 GB of RAM. The scripts for article pre-processing were implemented in Perl. The statistical analysis was performed using R [42] and the function 'lda' of package MASS [32]. The Andrews' curves were build using the R function "Andrews.curves" developed by Shigenobu AOKI[6]. The SVM model was build using the libsvm library [43]. The results of the classification using SVM were processed using R.

3. Experimentation

The experimentation followed the methodology described in the previous section and to better evaluate the performance of LDA, we had to exclude from Phase I (document retrieval and representation) as much bias as possible and at the same time compare LDA on a dataset that has already been used and performed well. Therefore, we had chosen the GO codes and the keywords proposed in [8] and kept the particular keywords that resulted in adequate number of articles for both training and testing. Our search and retrieval from PubMed has showed that for 9 of the GO codes given in [8] produced few results (ranging from 1 to 49), so if used they could be misleading for the training. Therefore, we have used in our experimentation 12 out of the 21 (used in [8]) GO codes, each of which has resulted in more than 175 articles. This was necessary, since for our experiments we used randomly drawn hold-out samples for validation, and therefore there was need to achieve representation of each GO code in all samples.

The first phase (document retrieval processing and conversion) resulted in a large number of vectors with $p = 1642$ variables representing words grouped into $k = 12$ GO codes listed in Table 1. These vectors were used to form the training and the test dataset.

[6] http://aoki2.si.gunma-u.ac.jp/R/Andrews.html.

3.1. *Dataset description*

The vectors created were divided into training and test data set according to their publication date. Specifically, the training dataset contained 10,485 articles published up to 1999 while the test dataset contained 10,706 articles between 2000 and 2004.

For the training dataset, there were 644 articles common for two different GO codes, 53 articles common for three different GO codes and six articles common to four GO codes. These were all excluded from the analysis since a basic assumption for LDA is that the groups are distinguished. Therefore, only 9014 articles were processed, each one corresponding to a unique GO code. Also, five articles had all of their vector elements equal to zero and were also excluded from the analysis. Finally, 9009 vectors remained in the training dataset. For the test dataset, a similar procedure resulted in 8225 articles.

3.2. *Results*

The methods described in Section 2 were applied in the datasets with special emphasis given on the evaluation of the discrimination and classification results of the LDA models. The main issues examined were the ability of the model to discriminate the groups of the training dataset and to correctly classify new cases. For better readability of the graphs each GO code is labelled by a group number ranging from 1 to 12 (Table 1).

3.2.1. *Results for the test dataset*

Regarding the performance of the LDA model in the classification of the articles in the test dataset, the confusion matrix was constructed (Table 2).

The metrics computed from the confusion matrix are shown in Table 3. The mean precision and recall for the test dataset is 77.2 and 74.3%, respectively, while the F-measure is 75.4%. The accuracy is 77.31%.

We can also see that the F-measure which is indicative for the performance of classification is quite different among GO codes. For example, group 1 (autophagy) has the largest 92.3% F-measure, while group 3 (cell proliferation) has the lowest, 32.1%. A possible explanation for this poor performance is that the term 'cell proliferation' defined as: "The multiplication or reproduction of cells, resulting in the rapid expansion of a cell population" [2] is highly correlated to other GO codes in the study, for example 'oncogenesis' and 'cell death'.

Another issue that could explain different degree of performance of the model among GO codes is the position of the GO term in the GO hierarchy. The "cell proliferation" category is very general and this has an effect on the performance of the model. The granularity of the GO codes is also mentioned in [8] as an issue affecting the correct classification. This problem deserves further investigation and could be addressed by enhancing the classification model with information from the GO hierarchy.

An important remark regarding the classification by LDA is that a new case is classified in a group according to the probabilities computed for all groups. The group that has the highest probability is assigned to the new case. After examination of the misclassified articles, we noted that the correct GO cate-

Table 2 – The confusion matrix with misclassifications for the test dataset

Actual group	Predicted group												Actual size
	1	2	3	4	5	6	7	8	9	10	11	12	
1	144	1	0	0	0	0	0	1	3	2	1	9	161
2	0	515	55	1	9	10	28	48	39	9	50	50	814
3	0	16	55	0	3	0	2	7	2	3	14	10	112
4	0	4	1	101	0	1	2	1	2	5	0	6	123
5	0	19	8	4	426	1	20	10	12	14	1	48	563
6	0	19	0	11	3	430	5	0	6	4	1	11	490
7	1	22	6	2	29	0	878	24	48	36	18	48	1112
8	1	36	32	0	1	0	21	397	24	6	32	52	602
9	2	39	10	3	6	8	66	24	806	11	21	65	1061
10	1	1	0	2	1	0	0	0	0	240	1	1	247
11	1	32	15	4	4	0	16	34	24	12	1289	59	1490
12	1	37	49	9	39	2	45	71	22	38	59	1078	1450
Predict size	151	741	231	137	521	452	1083	617	988	380	1487	1437	8225

Table 3 – Performance metrics for the test dataset

Group	Recall (%)	Precision (%)	*F*-measure (%)
1	89.4	95.4	92.3
2	63.3	69.5	66.3
3	49.1	23.8	32.1
4	82.1	73.7	77.7
5	75.7	81.8	78.6
6	87.8	95.1	91.3
7	89	81.1	84.9
8	66	64.3	65.1
9	76	81.6	78.7
10	97.2	63.2	76.6
11	86.5	86.7	86.6
12	74.3	75	74.6
Mean	77.2	74.3	75.4

gory was included most of the time between the four groups with the highest probabilities. This is shown in Table 4. For the test dataset 1020 articles that were misclassified using the first highest probability, could be assigned to the correct GO category using the second (rank 2) highest probability and so on. The practical importance of this remark is that in cases where the end user feels uncertainty regarding the final classification, he may take into account the lower-rank probabilities in order to figure out some alternative classification.

Additionally an important result of the LDA model is the computation of new variables, called linear discriminant (LD) variables, from linear transformations of the original ones. This offers a significant space reduction since the original 1642 variables are transformed to only 11 for our data (LD1–LD11). It is also important to notice in the analysis that LD variables are ranked according to their contribution in the discrimination of the groups. So, the first (LD1) has the most significant contribution for the discrimination between groups, the second (LD2) has the second most significant contribution, and so on [32]. This is evident from the eigenvalues returned as output from LDA (Table 5). The eigenvalues show the ratio of the between- and within-group standard deviations of the LD variables.

After obtaining the new variables LD1–LD11, we can now use them for the representation of the data by the graphical methods described in Section 2. Graphs like boxplots and scatterplots have specific practical importance: they can show how well the LDA model discriminates the groups and classifies the new cases. The Andrew's curves have a more interesting utility: the groups are provided with visual representation, i.e. each GO code gets a shape in the form of a curve. The same happens for the document vectors. So, by visual inspection of the shapes the user can identify resemblances between new documents and GO codes. Next, we give only some indicative examples of the graphical methods together with their appropriate interpretation.

The classification power of the LDA model can be assessed using boxplots, which portray how each LD variable is distributed in different GO codes. For example, Figs. 2 and 3

Table 4 – The number of articles assigned to the correct GO category based on the first and lower rank probabilities

Post probability rank	1	2	3	4	5	6	7	8	9	10	11	12
Number of articles	6359	1020	367	158	97	83	41	46	26	11	1	16

Table 5 – Eigenvalues for each of the 11 LD variables

LD	1	2	3	4	5	6	7	8	9	10	11
Eigenvalue	85.53	69.74	68.38	56.02	45.57	41.6	40.34	37.18	35.79	32.03	25.28

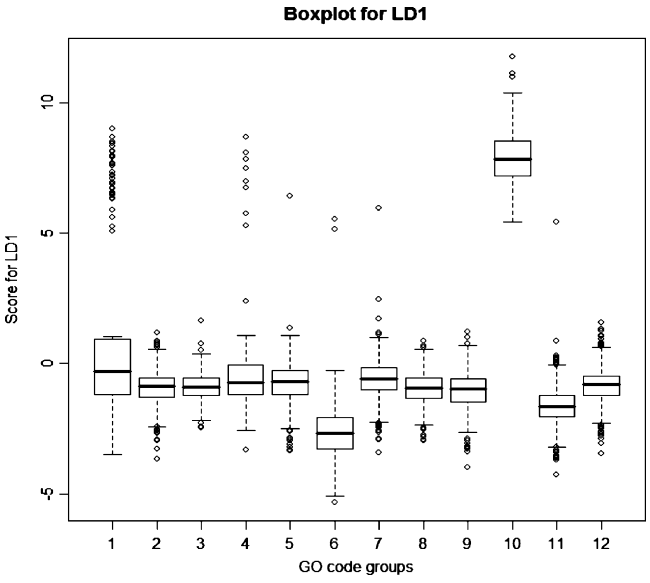

Fig. 2 – Boxplot of the articles scores for LD1.

show, respectively, the distributions of LD1 and LD3 of the test dataset over all GO categories. The obvious inference from these figures is that LD1 alone can discriminate very well the 10th category while LD3 clearly discriminates the first group. This relatively simple observation can provide the user with even more significant information regarding the classification potentiality of some keywords. This can be inferred by examining the weights assigned to the document terms by LDA for computing the LD variables. In our example, we can easily see that for the calculation of LD1, the terms "transport" and "golgi" have the largest weight, i.e. their contribution to LD1 is most significant. These terms are relevant to proteins with transport function, described by the 10th GO code. Similarly, the most important terms for LD3 are 'autophagy' and

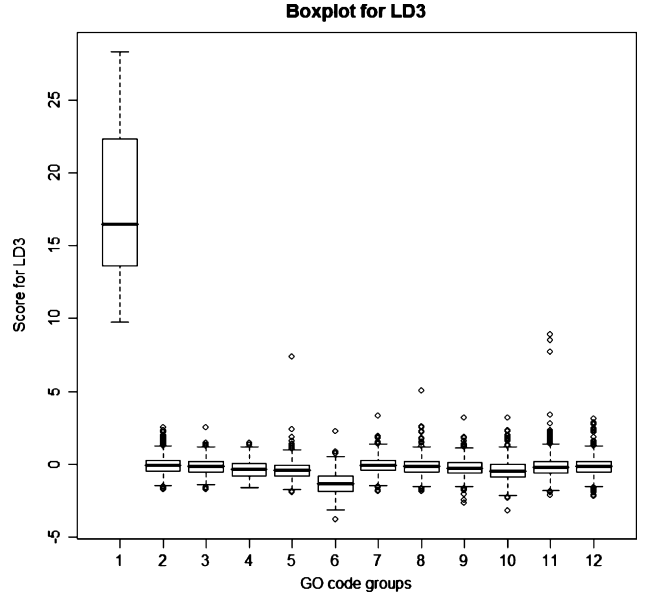

Fig. 3 – Boxplot of articles scores for LD3.

'lysosom',[7] which are characteristic for 'Autophagy' GO category.

Fig. 4 shows a matrix of scatterplots containing the centroids of the 12 GO categories with respect to the first four LD variables (LD1–4) in the test dataset. We can see that the combination of the first two LD variables discriminates clearly GO categories 1, 6 and 10. The addition of LD3 and LD4 enhances further the discrimination towards GO categories 11, 4, and 5. Such scatterplots are valuable in recognising which groups are discriminated better and which are more difficult to be distinguished.

Regarding Andrew's curves, each document can be represented by a single curve based on the 11 LD variables. Furthermore, we can represent by a single curve the whole GO group using the centroids corresponding to the LD variables of the training dataset. So, it is possible to compare the "document curve" with the "group curves" in order to classify visually the document to the category with the most similar curve. Of course, the different shapes of the group curves are indication of the classification efficiency of the LDA model and can also be used for validation of the model. For example, Fig. 5(a) shows the group curve of GO code 1 computed from the training dataset and Fig. 5(b) the curves of all the documents assigned to GO code 1 in the test dataset. In Fig. 6 all group curves are shown. We can clearly see that, in general, (a) the group curves are different and therefore well-discriminated; and (b) the document curves have strong similarity with the corresponding group.

3.2.2. *Results of the hold-out samples*

In order to assess the sensitivity of the model's classification robustness on changes of the training number of documents, we conducted repeated experiments with hold-out samples [38]. Three different sizes of hold-out samples were constructed containing 10, 20 and 30% of the total number of vectors of the training sample. For each size, a hold-out sample was drawn 10 different times, each time from different articles randomly selected. Each time the remaining training dataset was used for constructing the LDA model and the respective hold-out sample was used for validation. Then the performance of the ten different models based on the accuracy of classification of the hold-out sample was averaged, so as to have statistically robust results for the performance of the classification model.

Table 6 shows that the accuracy of the discriminant model for each of the three hold-out sample experiments is above 80% (Column 4) without significant changes when the size of the hold-out samples is growing (Column 2). This shows that the model is fairly stable and is not affected much by the loss in the number of training articles.

3.3. *Comparing LDA with SVM*

As previously mentioned SVM has been already applied for gene annotation in [19] and seems to outperform the maximum entropy method from [8].

[7] 'Lysosom' and not 'lysosome' due to stemming.

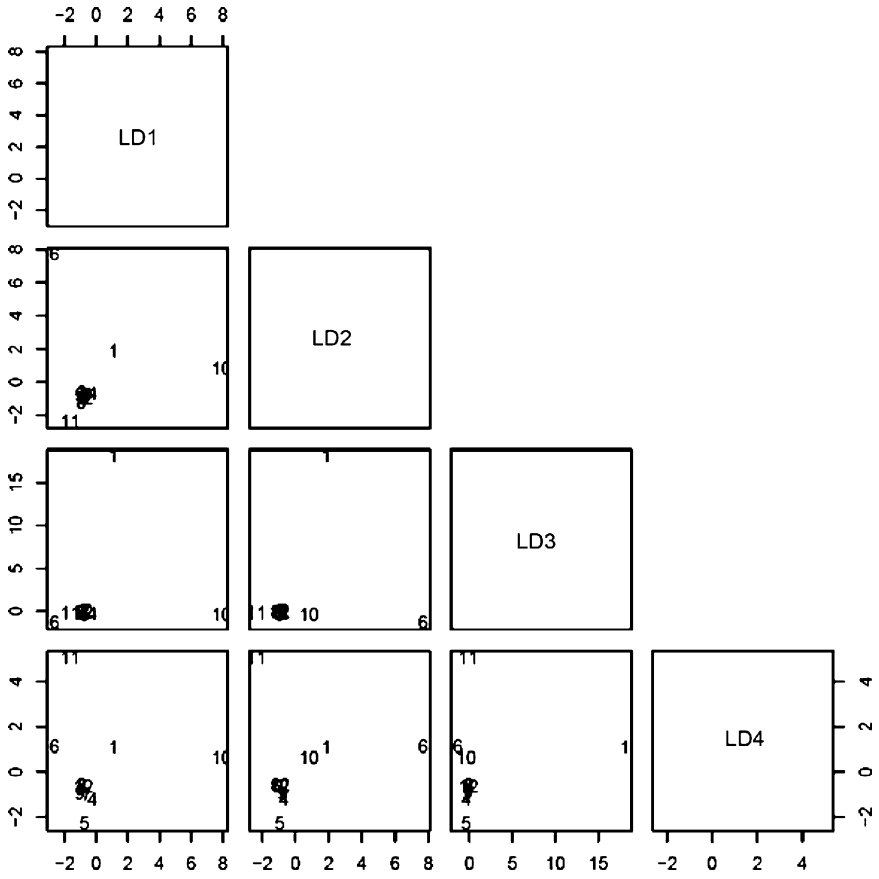

Fig. 4 – The centroids for each GO category of the test dataset for LD1–LD4.

As for LDA, the same training dataset was used for training the SVM model, whereas the same test dataset was used for measuring the performance of the model. A confusion matrix was created (Table 7) and recall, precision and *F*-measure were calculated for evaluating the results (Table 8).

Since we use the simplest form of LDA by building linear models, the kernel used in SVM was also linear. A general

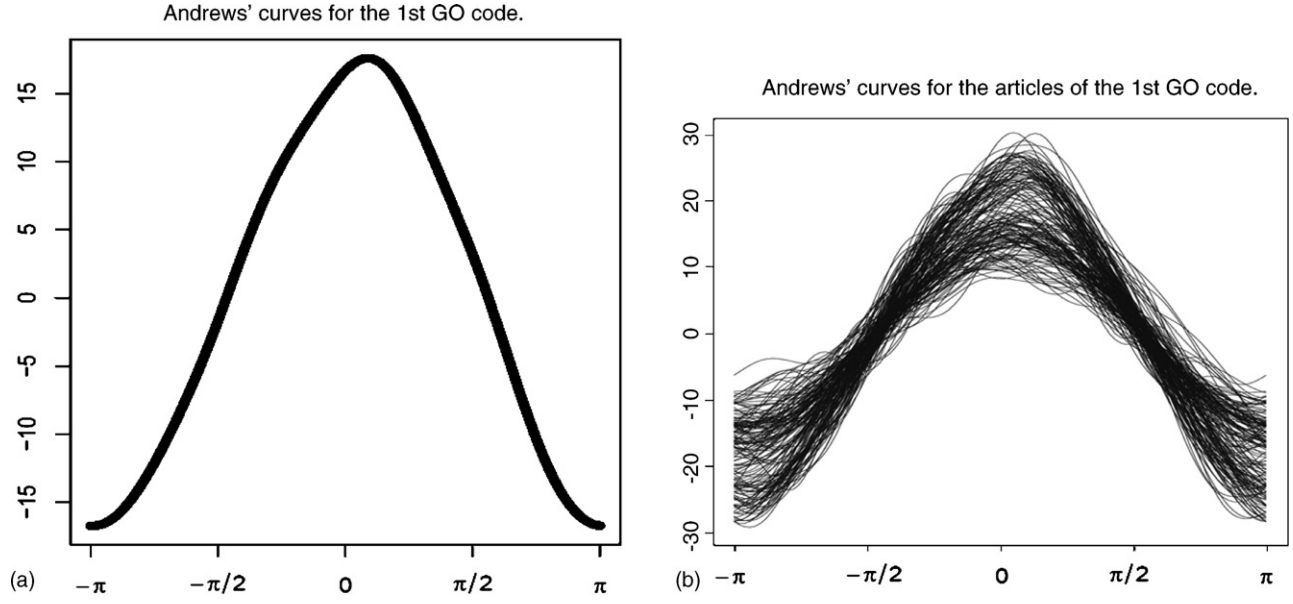

Fig. 5 – Andrews' curves for the first GO code (a) the group curve of the training dataset and (b) the document curves of the test dataset.

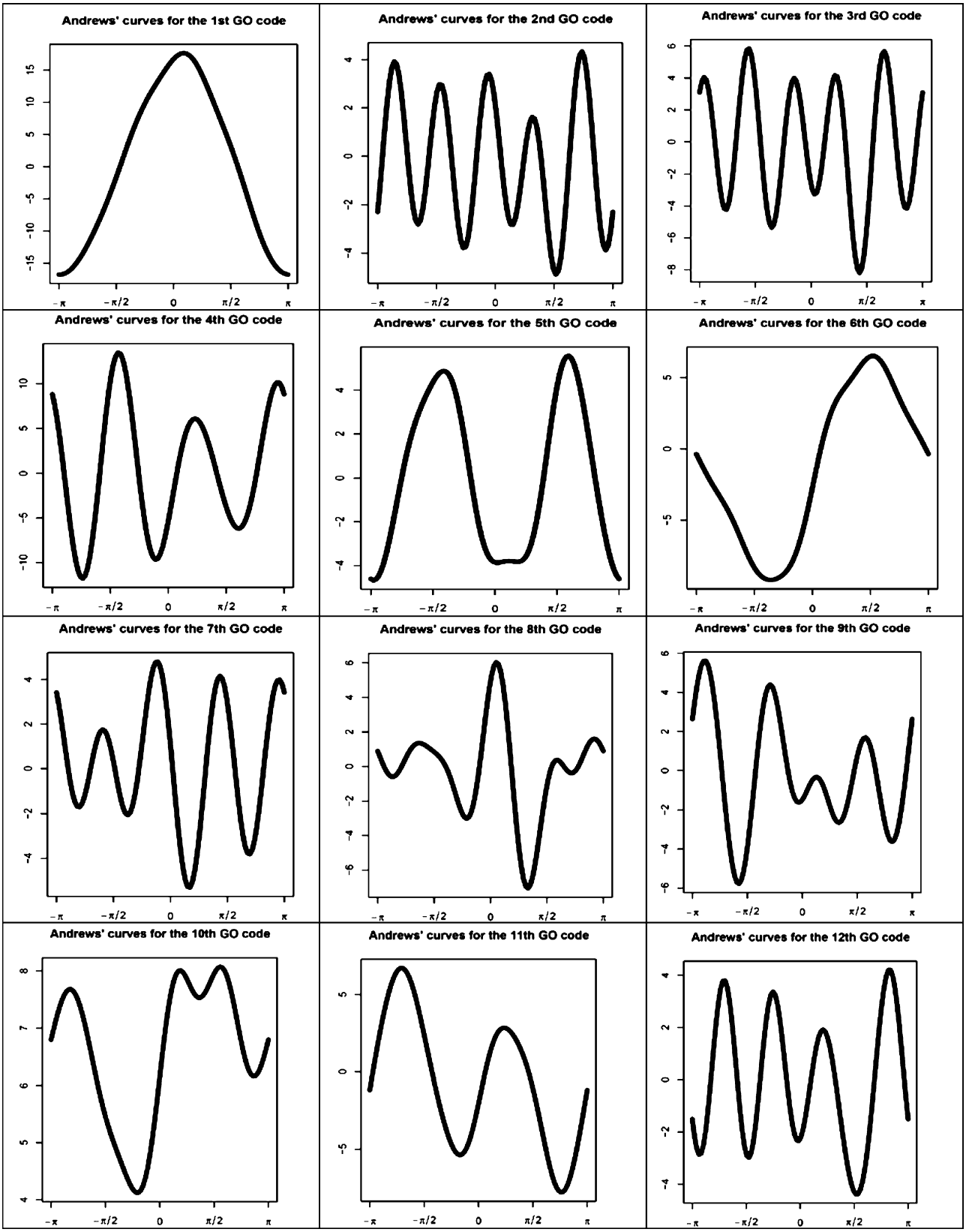

Fig. 6 – All group curves from the training dataset.

Table 6 – Results of the experiments with the hold-out samples

Experiment	Hold-out sample	Training dataset (remaining articles)	Accuracy	Standard deviation
Test 1	909 (10%)	8108 (90%)	81.90%	0.013
Test 2	1802 (20%)	7207 (80%)	81.87%	0.0076
Test 3	2703 (30%)	6306 (70%)	80.81%	0.0044

Table 7 – Confusion matrix for the test 2000–2004 dataset using SVM

Actual group	Predicted group												Actual size
	1	2	3	4	5	6	7	8	9	10	11	12	
1	104	0	0	2	3	1	1	3	2	16	18	11	161
2	1	443	51	4	26	35	36	73	65	5	31	54	814
3	0	15	54	0	3	0	0	16	2	1	10	11	112
4	0	1	1	88	5	1	7	3	3	5	2	7	123
5	0	14	8	8	404	4	26	22	25	9	4	39	563
6	2	12	3	9	6	420	3	6	14	3	2	10	490
7	3	30	8	0	60	3	733	90	89	31	20	45	1112
8	2	28	21	3	8	0	10	420	28	3	31	48	602
9	2	27	12	11	23	17	82	43	771	5	26	42	1061
10	0	0	0	1	3	0	11	1	3	221	2	5	247
11	1	39	16	3	9	3	19	59	52	8	1237	44	1490
12	3	47	29	17	38	3	71	89	45	24	58	1026	1450
Predict size	151	741	231	137	521	452	1083	617	988	380	1487	1437	8225

Table 8 – Performance of the classification using SVM

Group	Recall (%)	Precision (%)	F-measure (%)
1	64.6 (89.4%)	88.1 (95.4%)	74.6 (92.3%)
2	53.2 (63.3%)	67.0 (69.5%)	59.3 (66.3%)
3	48.2 (49.1%)	26.6 (23.8%)	34.3 (32.1%)
4	71.5 (82.1%)	60.3 (73.7%)	65.4 (77.7%)
5	71.6 (75.7%)	68.7 (81.8%)	70.2 (78.6%)
6	85.7 (87.8%)	86.2 (95.1%)	86.0 (91.3%)
7	65.9 (89%)	73.4 (81.1%)	69.4 (84.9%)
8	69.8 (66%)	51.0 (64.3%)	58.9 (65.1%)
9	72.7 (76%)	70.2 (81.6%)	71.4 (78.7%)
10	89.5 (97.2%)	66.8 (63.2%)	76.5 (76.6%)
11	83.0 (86.5%)	85.8 (86.7%)	84.4 (86.6%)
12	70.6 (74.3%)	76.5 (75%)	73.5 (74.6%)
Mean	70.1 (77.2%)	68.4 (74.3%)	68.7 (75.4%)

Inside the parentheses are the results from the LDA classification.

remark is that the performance is worse than that of LDA. Nevertheless, in some cases, like in the eighth group the recall is better than the one of the LDA model. Also, the third group has, as in LDA, the smallest *F*-measure as seen in Table 8, but it is better than the performance in LDA. Nevertheless, the overall *F*-measure for SVM is 68.7%, which is quite lower than LDA. The general conclusion is that LDA is competitive and can be compared to the performance of the classification from SVM. This is consistent with other works ([21,22]) that measure the performance of LDA classification models against SVM and other classification methods and conclude that LDA can perform very well and compete with newer methods.

4. Conclusion

We can conclude from the results of the experimentation that LDA can classify efficiently data from biological texts and therefore can be used to facilitate the process of assigning a GO category to a gene product, using information from published biological literature. LDA is a simple statistical procedure, based on linear models, that is proved to be competitive to other classifiers, like SVM ([21,22]). Furthermore, LDA performs transformations on the original data and produces fewer new variables that can provide better understanding of the documents datasets and especially of the groupings that are essential for gene annotation.

For our experimentation we used a small number of GO codes, specifically those for which we had a large number of documents assigned to each one of them. We have to emphasize that this decision was taken only for the experimentation procedure and has nothing to do with the classification accuracy. Our wish was to ensure that the randomly chosen hold-out samples would contain representatives of all the GO codes so as to avoid problems related to bias in the comparisons.

Summary points

What was already known on the topic?

- The function of a gene can be described with informative or predefined keywords or Bio-Ontology terms.
- Gene Ontology terms are well-suited for describing gene functions independently of the organism and the cell type.
- The computational methods used so far for assigning a function to a gene use as a source of information either sequence analysis or biomedical articles.
- Maximum entropy, naive Bayes, nearest-neighbor and support vector machines have been already used successfully for assigning a Gene Ontology code to a gene using the information in biomedical literature.

What did this study add to our knowledge?

- A well-known statistical method, Linear discriminant analysis, is exploited, as an alternative to previous methods, for functionally annotating genes.
- LDA can develop models that can be used for the generation of new variables which can offer better insights and reduce the dimensionality of a bio-text dataset.
- The graphical representation of the new variables facilitate the understanding and the evaluation of the classification results.
- A statistical framework is proposed both for building classification models and for evaluating them. The evaluation is achieved by a combination of performance metrics and graphical representations.

In real situations, one has a standard (and very large) corpus of documents corresponding to many GO codes and wishes to classify a new document to one of them. This is done automatically by the LDA model no matter how small is the number of documents of each group. Note that LDA in the estimation of probabilities for classification takes into account the size of each group. Of course when the number of GO codes is increasing, the graphical methods are getting difficult to interpret. In these cases we can exploit the LD variables, which are ranked according to their discriminating importance and work with a smaller number of them, the most important ones.

An important aspect that could also be addressed in the future is the information contained inside articles for each GO category and the use of better selection methods of the articles. The selection of articles is a crucial step in the methodologies of supervised classification models, since it defines the quality of the training dataset. Ideally for our method the articles should contain information relevant only to a specific GO category. An interesting future research direction is the extension of the proposed methodology for handling simultaneously several GO codes, which would better reflect the way the GO terms are manually assigned today. There exist several multivariate statistical methods ([36,37]), which could be used for handling simultaneously multiple GO terms, i.e. methods based on multi-response regression.

Acknowledgment

The authors thank the anonymous referees for their valuable comments and suggestions, which contributed in considerably improving the paper's content, presentation and readability.

REFERENCES

[1] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. Matese, J. Richardson, M. Ringwald, G. Rubin, G. Sherlock, Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, Nat. Genet. 25 (1) (2000) 25–29, http://dx.doi.org/10.1038/75556.

[2] G.O. Consortium, Creating the gene ontology resource: design and implementation, Genome Res. 11 (8) (2001) 1425–1433.

[3] P.D. Karp, M. Riley, M. Saier, I.T. Paulsen, J. Collado-Vides, S.M. Paley, A. Pellegrini-Toole, sar Bonavides, S. Gama-Castro, The EcoCyc database, Nucleic Acids Res. 30 (1) (2002) 56–58.

[4] E. Camon, M. Magrane, D. Barrell, D. Binns, W. Fleischmann, P. Kersey, N. Mulder, T. Oinn, J. Maslen, A. Cox, R. Apweiler, The Gene Ontology Annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro, Genome Res. 13 (4) (2003) 662–672, http://dx.doi.org/10.1101/gr.461403.

[5] 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, S. Weng, D. Botstein, J.M. Cherry, Saccharomyces Genome Database (SGD) provides secondary gene annotation using the Gene Ontology (GO), Nucleic Acids Res. 30 (1) (2002) 69–72.

[6] H. Xie, A. Wasserman, Z. Levine, A. Novik, V. Grebinskiy, A. Shoshan, L. Mintz, Largescale protein annotation through gene ontology, Genome Res. 12 (5) (2002) 785–794, http://dx.doi.org/10.1101/gr.86902.

[7] PubMed, National Center for Biotechnology Information (NCBI), National Library of Medicine (NLM), National Institute of Health (NIH), http://www.pubmed.com.

[8] S. Raychaudhuri, J.T. Chang, P.D. Sutphin, R.B. Altman, Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature, Genome Res. 12 (1) (2002) 203–214, http://dx.doi.org/10.1101/gr.199701.

[9] D.L. Wheeler, D.M. Church, R. Edgar, S. Federhen, W. Helmberg, T.L. Madden, J.U. Pontius, G.D. Schuler, L.M. Schriml, E. Sequeira, T.O. Suzek, T.A. Tatusova, L. Wagner, Database resources of the National Center for Biotechnology Information: update, Nucleic Acids Res. 32 (2004) D35–D40 (Database issue) http://dx.doi.org/10.1093/nar/gkh073.

[10] B. de Bruijn, J. Martin, Getting to the (c)ore of knowledge: mining biomedical literature, Int. J. Med. Inform. 67 (1–3) (2002) 7–18.

[11] S. Khan, G. Situ, K. Decker, C.J. Schmidt, GoFigure: automated Gene Ontology annotation, Bioinformatics 19 (18) (2003) 2484–2485.

[12] D.M. Martin, M. Berriman, G.J. Barton, GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes, BMC Bioinform. 5 (2004) 178.

[13] M. Andrade, A. Valencia, Automatic annotation for biological sequences by extraction of keywords from MEDLINE

abstracts. Development of a prototype system, Proc. Int. Conf. Intell. Syst. Mol. Biol. 5 (1997) 25–32.
[14] H. Shatkay, S. Edwards, W. Wilbur, M. Boguski, Genes, themes and microarrays: using information retrieval for large-scale gene analysis, Proc. Int. Conf. Intell. Syst. Mol. Biol. 8 (2000) 317–328.
[15] J. Tamames, C. Ouzounis, G. Casari, C. Sander, A. Valencia, EUCLID: automatic classification of proteins in functional classes by their database annotations, Bioinformatics 14 (6) (1998) 542–543.
[16] F. Eisenhaber, P. Bork, Evaluation of human-readable annotation in biomolecular sequence databases with biological rule libraries, Bioinformatics 15 (7–8) (1999) 528–535.
[17] A. Vinayagam, R. Knig, J. Moormann, F. Schubert, R. Eils, K.H. Glatting, S. Suhai, Applying support vector machines for Gene Ontology based gene function prediction, BMC Bioinform. 5 (1) (2004) 116, http://dx.doi.org/10.1186/1471-2105-5–116.
[18] S. Hennig, D. Groth, H. Lehrach, Automated Gene Ontology annotation for anonymous sequence data, Nucleic Acids Res. 31 (13) (2003) 3712–3715.
[19] T. Izumitani, H. Taira, H. Kazawa, E. Maeda, Assigning gene ontology categories (go) to yeast genes using text-based supervised learning methods, in: Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference (CSB 2004), 2004, pp. 503–504.
[20] K.R. Christie, S. Weng, R. Balakrishnan, M.C. Costanzo, K. Dolinski, S.S. Dwight, S.R. Engel, B. Feierbach, D.G. Fisk, J.E. Hirschman, E.L. Hong, L. Issel-Tarver, R. Nash, A. Sethuraman, B. Starr, C.L. Theesfeld, R. Andrada, G. Binkley, Q. Dong, C. Lane, M. Schroeder, D. Botstein, J.M. Cherry, Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms., Nucleic Acids Res. 32 (2004) D311–D314 (Database issue) http://dx.doi.org/10.1093/nar/gkh033.
[21] T. Lim, W. Loh, Y. Shih, A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, Machine Learn. 40 (3) (2000) 203–228.
[22] D. Meyer, F. Leisch, K. Hornik, Benchmarking support vector machines, Tech. Rep. 78, Vienna University of Economics and Business Administration in cooperation with University of Vienna, Vienna University of Technology, 2002.
[23] W.J. Krzanowski, Principles of Multivariate Analysis. A User's Perspective, Oxford Science Publications, 1988.
[24] C.A. Bean, R. Green (Eds.), Relationships in the Organization of Knowledge, Kluwer Academic Publishers, NY, 2001, pp. 171–184.
[25] C. Perez-Iratxeta, P. Bork, M. Andrade, Association of genes to genetically inherited diseases using data mining, Nat. Genet. 31 (2002) 316–319.
[26] G. Salton, Automatic text analysis, Science 168 (1970) 335–343.
[27] H. Shatkay, R. Feldman, Mining the biomedical literature in the genomic era: an overview, J. Comput. Biol. 10 (6) (2003) 821–855, http://dx.doi.org/10.1089/106652703322756104.
[28] C.D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing, The MIT Press, 1999.
[29] J.M. Daniel Jurafsky, Speech and natural language processing, Alan Apt., 2000, ISBN 0-13-095069-6.
[30] C. Paice, Another stemmer, SIGIR Forum 24 (3) (1990) 56–61.
[31] J. Zobel, A.M. offat, Exploring the similarity space, SIGIR Forum 32 (1) (1998) 18–34.
[32] W.N. Venables, B. D. Ripley, Modern Applied Statistics with S. Fourth Edition, fourth ed., Springer, 2002, ISBN 0-387-95457-0.
[33] C. Cortes, V. Vapnik, Support-vector networks, Machine Learn. 20 (1995) 273–297.
[34] P. Baldi, P. Frasconi, P. Smyth, Modeling the Internet and the Web, John Wiley and Sons Ltd, 2003.
[35] R.A. Fisher, The use of multiple measurements in taxonomic problem, Ann. Eugenices 7 (1936) 179–188.
[36] J. Lattin, D. Carroll, P. Green, Analyzing multivariate data, Curt. Hinrichs (2003).
[37] B.G. Tabachnick, L.S. Fidell, Using Mutlivariate Statistics, Allyn and Bacon, 2001, ISBN 0-321-05677-9.
[38] J.F. Hair, R.E. Anderson, R.L. Tatham, W.C. Black, Mutlivariate Data Analysis, fifth ed., Prentice Hall PTR, 1998, ISBN 0-13-930587-4.
[39] W.S. Cleveland, Visualizing Data, Hobart Press, 1993.
[40] D.F. Andrews, Plots of high-dimensional data, Biometrics 28 (1972) 125–136.
[41] C.J. Van Rijsbergen, Information Retrieval, second ed., Department of Computer Science, University of Glasgow, 1979.
[42] R Development Core Team, R: a language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, 2004, ISBN 3-900051-00-3, URL http://www.R-project.org.
[43] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.