Journal of Biomedical Informatics 46 (2013) 1044–1059

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

A simulation to analyze feature selection methods utilizing gene ontology for gene expression classification

Christopher E. Gillies [a], Mohammad-Reza Siadat [a,*], Nilesh V. Patel [a], George D. Wilson [b]

[a] Dept. of Computer Science and Engineering, Oakland University, 2200 N Squirrel Rd, Rochester, MI 48309, United States
[b] Beaumont Health System, 3601 W. Thirteen Mile Rd, Royal Oak, MI 48073, United States

ARTICLE INFO

Article history:
Received 26 September 2012
Accepted 21 July 2013
Available online 25 July 2013

Keywords:
Data mining
Gene expression
Cancer classification
Gene ontology
Semantic similarity
Feature evaluation and selection

ABSTRACT

Gene expression profile classification is a pivotal research domain assisting in the transformation from traditional to personalized medicine. A major challenge associated with gene expression data classification is the small number of samples relative to the large number of genes. To address this problem, researchers have devised various feature selection algorithms to reduce the number of genes. Recent studies have been experimenting with the use of semantic similarity between genes in Gene Ontology (GO) as a method to improve feature selection. While there are few studies that discuss *how* to use GO for feature selection, there is no simulation study that addresses *when* to use GO-based feature selection. To investigate this, we developed a novel simulation, which generates binary class datasets, where the differentially expressed genes between two classes have some underlying relationship in GO. This allows us to investigate the effects of various factors such as the relative connectedness of the underlying genes in GO, the mean magnitude of separation between differentially expressed genes denoted by δ, and the number of training samples. Our simulation results suggest that the connectedness in GO of the differentially expressed genes for a biological condition is the primary factor for determining the efficacy of GO-based feature selection. In particular, as the connectedness of differentially expressed genes increases, the classification accuracy improvement increases. To quantify this notion of connectedness, we defined a measure called Biological Condition Annotation Level $BCAL(G)$, where G is a graph of differentially expressed genes. Our main conclusions with respect to GO-based feature selection are the following: (1) it increases classification accuracy when $BCAL(G) \geqslant 0.696$; (2) it decreases classification accuracy when $BCAL(G) \leqslant 0.389$; (3) it provides marginal accuracy improvement when $0.389 < BCAL(G) < 0.696$ and $\delta < 1$; (4) as the number of genes in a biological condition increases beyond 50 and $\delta \geqslant 0.7$, the improvement from GO-based feature selection decreases; and (5) we recommend not using GO-based feature selection when a biological condition has less than ten genes. Our results are derived from datasets preprocessed using RMA (Robust Multi-array Average), cases where δ is between 0.3 and 2.5, and training sample sizes between 20 and 200, therefore our conclusions are limited to these specifications. Overall, this simulation is innovative and addresses the question of *when* SoFoCles-style feature selection should be used for classification instead of statistical-based ranking measures.

1. Introduction

A major transformation is occurring within the health-care community. Instead of applying general treatments to diverse patient populations, therapies are being customized to patient subpopulations based on their gene expression profiles. This new form of medical practice is known as personalized medicine. Gene expression profile classification is subfield of bioinformatics that is aiding in the transformation to personalized medicine. Gene expression profiling is an important tool for personalized medicine because it allows biomedical researchers to discover biomarkers. There are two types of biomarkers, prognostic biomarkers and predictive biomarkers. Prognostic biomarkers allow clinicians to discern which patients to treat, while predictive biomarkers elucidate a treatment's effectiveness for a patient [1]. For gene expression studies, a biomarker is represented by the expression of a gene or set of genes under a certain physiological condition. To give the reader some insight regarding biomarkers, suppose some treatment t_{reat} has been shown to be effective for patients having some biomarker b in physiological situation s_{it}. Now suppose we have a patient p with physiological situation s_{it}, we want to say: if patient p with physiological situation s_{it} has biomarker b then apply treatment t_{reat} to patient p. In this example, b is a

* Corresponding author.
E-mail addresses: cegillie@oakland.edu (C.E. Gillies), siadat@oakland.edu (M.-R. Siadat), npatel@oakland.edu (N.V. Patel), george.wilson@beaumont.edu (G.D. Wilson).

1532-0464/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jbi.2013.07.008

predictive biomarker. There are many applications of gene expression profile classification including: tumor class discrimination, prediction of clinical outcome based on treatments and detection of previously unknown sub-patterns [2]. Gene expression profiles have been traditionally collected using DNA microarrays, however, recent studies are also using RNA-seq [3]. In its simplest form, the gene expression classification problem compares two classes: (1) a control class and (2) an experimental class. Dubitzky et al. describe nine steps involved with microarray data analysis: (1) identify scientific aims; (2) design experiment; (3) design/make or acquire microarray; (4) hybridize and scan microarray; (5) analyze resulting image; (6) derive data matrix; (7) pre-process data matrix; (8) analyze and model; and (9) interpret and validate results [4].

In this paper, we focus on steps seven and eight of microarray data analysis. Step seven has a few subtasks such as missing value computation, normalization, transformation and feature selection. Within step seven, we are most interested in feature selection. The ultimate goal of feature selection, also known as gene selection, is to reduce the dimensionality of the problem and identify potential biomarkers. Feature selection is supremely important because a gene expression profile has thousands of values associated with it, and fitting a classifier with exceptionally high dimension leads to the curse of dimensionality. Adding to the problem is the fact that there are usually only tens or hundreds of gene expression profiles to use as training examples. With regards to step eight, many different classifiers such as linear discriminant analysis (LDA) [5], diagonal linear discriminant analysis (DLDA) [5], weighted voting (WV) [6], k-nearest neighbor (KNN) [7] and, support vector machines (SVM) [8] have been applied on gene expression profiles [9].

To frame the problem in a more mathematical context, let's define the training set to be $T \in \mathbb{R}^{m \times n}$ where m is the number of genes, n is the number of biospecimens analyzed and $\mathbb{R}^{m \times n}$ refers to a m by n dimensional real number space. A column of the set T represents the gene expression profile of a biospecimen; a row represents the expression levels for a single gene across all biospecimens. The primary objective of feature selection is to find $T' \in \mathbb{R}^{d \times n}$ where $T' \subset T$ and $d < m$ such that training a classifier C on T' yields higher generalized classification accuracy than training on T.

Given the importance of discovering biomarkers, one should not be surprised to find a vast amount of literature devising feature selection algorithms. There are three common approaches to feature selection: filter, wrapper, and embedded techniques [10]. To understand how feature selection algorithms are classified using this scheme, it is useful to envision these processes as they occur along a time-line with respect to training a classifier. Specifically, filtering occurs before classifier training, embedded selection occurs during classifier training and wrappers are applied after classifier training. Filtering techniques typically rank genes by some statistical metric and then remove all genes that fall below a user-defined threshold. Wrapper methods attempt to find an optimal subset of genes that achieve high accuracy. These methods are called wrappers because they encapsulate a classifier and call the classifier as a subroutine. Table 1 lists some feature selection techniques.

Table 1
Examples of feature selection techniques.

Technique	Type	Publication
Signal-to-noise ratio	Filter	[6]
t-Statistics	Filter	[11]
ANOVA	Filter	[12]
Wilcoxon rank-sum	Filter	[13]
BLOCK.FS	Wrapper	[14]
Multiple SVM-RFE	Wrapper	[15]
Integer-coded genetic algorithm	Wrapper	[16]
Genetic programming	Embedded	[17]
Multiple-filter–multiple-wrapper	Combination	[9]

The previously mentioned techniques discover important genes by comparing statistical properties of a dataset, and do not include domain specific biological knowledge into the selection. Recently, some researchers have been investigating whether or not prior knowledge could improve feature selection for gene expression data classification. The rationale for these investigations is based on the following inductive argument: (1) gene expression data has small sample sizes, so the identification of important genes is difficult; (2) there are large biomedical knowledge-bases such as Gene Ontology (GO) [18] and Gene Ontology Annotation (GOA) [19] that describe gene relationships; (3) there appears to be some correlation between gene expression data and semantic similarity between terms in GO [20,21]; (4) there has been success by incorporating prior knowledge in other pattern recognition tasks; therefore, it seems possible that incorporating prior knowledge into feature selection techniques will improve biomarker identification and classification accuracy for gene expression data.

Further support for feature selection techniques that incorporate prior knowledge can be found in the success of enrichment analysis tools. The purpose of enrichment analysis tools is to assist with the interpretation of a list of relevant genes from data generated using high-throughput technologies like microarrays. Huang et al. mention that these tools are built on the following assumption: if a biological process is not functioning properly, then genes involved in this biological process will have a higher likelihood to be relevant [22]. The goal of these enrichment analysis tools is to find biological processes that best describe a user-specified list of relevant genes. Some of these enrichment analysis tools discover relevant biological terms by comparing a GO term's coverage among the list of relevant genes to its coverage among all genes. The difference between feature selection methods and enrichment analysis tools is that feature selection methods build a list of relevant genes, where as enrichment analysis tools assist with the interpretation of a list of relevant genes.

Some examples of enrichment analysis tools are Onto-Express [23], MaPPFinder [24], GOMiner [25], DAVID [26], EASE [27], GeneMerge [28], and FuncAssociate [29]. Refinements to enrichment analysis tools using information theory can be found in [30]. Other enrichment analysis tools, do not require a list of relevant genes, instead they work on all the genes. An example is Gene Set Enrichment Analysis (GSEA) discussed in [31,32]. GSEA works on a ranked list of genes that are correlated with a phenotype. GSEA tries to discover functional annotations such as GO terms that are either up-regulated or down-regulated relative to a control group. This allows for functional annotation-level analysis.

Extensions of functional annotation-level analysis methods are found in signatures of pathway deregulation in tumors [33], Condition-Responsive Genes (CORGs) [34] and the Functional Analysis of Individual Microarray Expression (FAIME) profiles [35]. Two other functional-level analysis methods aimed at the interpretation of high-throughput biological results are [36,37]. Functional-level analysis, similar to other enrichment analysis tools, are also used to interpret high-throughput results, however, methods like FAIME map gene expression values onto functional-level annotations such as GO terms. This mapping procedure allows pattern recognition tasks to be performed directly at the functional-level instead of at the gene-level. Feature selection methods, as discussed in this paper, select important genes at the gene-level.

SoFoCles [38] is a feature selection technique, which is based, in part, on Qi and Tang's method [39,40]. SoFoCles uses information from GO to improve statistical feature selection. In our paper, we refer to the enrichment of feature selection using GO as GO-based feature selection. The authors of SoFoCles show that GO-based

feature selection improves classification accuracy for two datasets. While these are impressive results, one important question is: Does this technique generalize to other datasets? If the answer to this is yes, another question immediately follows: Should GO always be used to enhance feature selection? In particular, under what conditions does GO-based feature selection lead to improved classification accuracy?

In this paper, we investigate *when* GO-based feature selection is effective. This differs from previous studies that investigate *how* to use GO effectively. To our knowledge, no existing study performs this analysis. Methods such as GO PaD [30] apply information theory to select *which* GO terms to use for the functional analysis and the interpretation of a list of genes selected from gene expression data. We assess *when* GO as a whole should be used for feature selection for gene expression profile classification. In summary, we assess *when* GO should be used to select genes and GO PaD assess *which* GO terms should be used to interpret the results of gene expression analysis. An important question related to our goal is whether to use real data or simulated data. If we used real data, it would be difficult to comprehend the underlying mechanisms that contribute to improved classification accuracy, and we would of had small sample sizes, thus it would be difficult to draw valid conclusions when comparing GO-based feature selection versus statistical feature selection. Based on this, we opted for a simulation-based approach. But, how does one generate synthetic data from GO? In order to answer this question, some background knowledge is required. For example, what exactly is GO? We answer questions like this and many others in Section 2. Once this foundation is in place, we move onto Section 3 where we unveil our methodology for simulation. Our preliminary study, [41], and the fact that GO and GOA define the relationships between genes, led us to hypothesize that the amount of accuracy improvement GO-based feature selection yields is proportional to the connectedness in GO of the differentially expressed genes characterizing a particular biological phenomenon. Also in this section, we define a measure to quantify the notion of connectedness. We present our results of utilizing this simulation to evaluate our hypothesis in Section 4. Finally, in Section 5, we summarize our work, and draw conclusions.

2. Background

Before we introduce our simulation, we must acquaint the reader with some fundamental concepts relating to our study. In this section, we begin by discussing GO, which is a bioinformatics resource containing a set of biological terms and the relationships between those terms. From here, we introduce Gene Ontology Annotation (GOA), which is a database that allows us to relate a gene's products to specific GO terms. Next, we introduce information content (IC), which allows us quantify the specificity, or how much information is contained within a term in GO. Semantic similarity builds on the idea of information content, and it allows us to compare the similarity of two terms in GO. These concepts constitute the foundation of our simulation methods.

2.1. Gene ontology and gene ontology annotation

An ontology is a "specification of conceptualization" [42], in practice an ontology has a set of terms and relations between these terms. GO was created by the Gene Ontology Consortium [18] to support the development of a controlled vocabulary which biologists could use to collaborate more precisely. In addition, GO allows researchers to compare gene function profiles between different species, which was extremely difficult prior to GO's existence [43]. The terms in GO are organized as a directed acyclic graph (DAG). In GO, two terms are related by either a "part_of" edge or an "is_a" edge. GO was designed to be species neutral to collectively describe biological concepts from multiple organisms. There are three disjoint DAGs in GO: cellular components, molecular functions and biological processes. The cellular component ontology "refers to the location inside or outside of the cell where a gene product is active" [43]. The molecular function ontology has terms that describe specific biochemical activities in the cell. A gene product or a set of gene products can perform a molecular function. A term in the biological process ontology "describes the objective which a gene or gene product contributes," and these terms usually represent a set of molecular functions with a well-defined beginning and end [43]. The structure of the biological process ontology suggests a possible filtering method for finding important genes. Suppose a gene is identified to be significant, a possible way to find other important genes would be to look for genes involved with the same or similar biological processes. To capitalize on an approach like this, there needs to be a way to map from genes to GO terms.

GO terms are annotated to gene products via the GOA [19] project. When a GO term is annotated to a gene product, the functional properties of the GO term are inherent to the gene product. There is a many-to-many relationship between GO terms and gene products. Each annotation has an evidence code associated with it. There are two broad categories of annotations: manual annotations and automatically assigned evidence codes. Biocurators read full text articles on resources such as PubMed and transfer information into GOA using GO terms [43]. The computational approach attempts to mimic manual biocuration. There are some evidence codes that are fully manual, experts verify others, and the only fully automatic evidence code is Inferred from Electronic Annotation (IEA). Since curators do not review this evidence code, it is considered less reliable than the manual evidence codes. IEA annotations are discovered by cross-referencing corresponding data from multiple biomedical-databases, and they are usually based off of the results from sequence alignments or scientific text mining [38]. In summary, GO is a controlled vocabulary that represents biological terms and the relationships between those terms, and GOA is a database that labels gene products with GO terms.

2.2. Information content and semantic similarity

Since GO terms are annotated to a gene's products, before any similarity value between two genes can be assigned, there first must be a way to quantify the similarity between two GO terms. And, prior to this quantification, there needs to be a technique to assign a numerical value to each term in GO. Within GO, there is a tendency for GO terms closer to the root to represent more general concepts and terms that are closer to the leaves to represent more specific concepts. Information content is a measure of specificity of a particular concept, which captures the amount of information intrinsic to each GO term. Terms pertaining to more specific concepts (e.g., induction of positive chemotaxis) have more information thus larger values, while those terms corresponding to more general concepts (e.g., biological process) have less information thus smaller values. The information content [38] of a GO term t can be expressed as:

$$IC(t) = -\log(p(t)) = -\log\left(\frac{n_t}{n_r}\right) = \log(n_r) - \log(n_t) \quad (1)$$

where $p(t)$ is the probability of t in GO, n_t is the count of the term and its descendants in GO, and n_r is the count of the root and its descendants. In GO, n_r is equal to the number of terms in the ontology, because the root is an ancestor of all other terms. Using this definition, a leaf in GO has maximal information content, and the

root has information content equal to zero. The information content is intrinsic to GO, thus there is no external corpus to compute information content [44]. Using this intrinsic information concept, a descendant of a term can be thought of as another occurrence of its ancestors. We used the convention that a term is a descendant and an ancestor of itself, because this was the default parameter setting in MATLAB,[1] the software platform used for this investigation. One can normalize the information content measure so it takes on values between zero and one by dividing by the information content of a leaf. Since leaves have maximal information content, this forces their normalized information content to one [38]. The mathematical derivation of this notion can be seen below.

$$IC(leaf) = -\log(p(leaf)) = -\log\left(\frac{1}{n_r}\right) \tag{2}$$

$$IC_{norm}(t) = \frac{IC(t)}{IC(leaf)} = \frac{\log\frac{n_t}{n_r}}{\log(\frac{1}{n_r})} = 1 - \frac{\log(n_t)}{\log(n_r)} \tag{3}$$

The concept of information content leads to numerous methods for calculating the semantic similarity between two GO terms: [45–47]. The idea of semantic similarity is to compute the amount of common information between two terms. Please see [48] for an a review of semantic similarity methods for biomedical ontologies. One measure for semantic similarity is Resnik, and this method works by assigning the semantic similarity between two terms to be the information content of their lowest common ancestor [45].

The mathematical formula of the normalized Resnik semantic similarity of two terms t_1 and t_2 is:

$$\text{R-}sim_{norm}(t_1, t_2) = \frac{\max_{t \in S(t_1,t_2)}[IC(t)]}{IC(leaf)} \tag{4}$$

where $S(t_1, t_2)$ is the common set of ancestors for t_1 and t_2.

The final prerequisite for computing the semantic similarity between two genes is a way to compare multiple terms simultaneously. This is required because in GOA genes can be annotated to a set of GO terms. With two genes, we can represent the similarity between the two sets of GO terms corresponding to the two genes as a matrix:

$$SIM(a, b) = \begin{bmatrix} sim_{1,1} & sim_{1,2} & \cdots & sim_{1,N_b} \\ \vdots & \vdots & \ddots & \vdots \\ sim_{N_a,1} & sim_{N_a,2} & \cdots & sim_{N_a,N_b} \end{bmatrix} \tag{5}$$

where N_a is the number of GO terms for gene a and N_b is the number of GO terms for gene b.

The similarity between gene a and gene b can be assigned by $Sim_{MAX}(a, b)$, which finds the maximum value of the matrix $SIM(a, b)$:

$$Sim_{MAX}(a, b) = \max_{i,j}(sim_{i,j}) \tag{6}$$

In this section, we introduced the concept of information content, which is a method to quantify the amount of information intrinsic to a GO term. We then discussed a method to compute the semantic similarity between two GO terms. This method assigns the semantic similarity between two GO terms to be the information content of their lowest common ancestor. This notion represents the amount of common information between these two terms. Finally, we mentioned an approach to compare the semantic similarity between two genes by finding the maximum similarity between the GO terms annotated to these genes.

3. Methods

The purpose of this section is to explain the methods used in our simulation. We first discuss how our simulation generates gene expression data from GO. We used version 1.1807 of GO, which we downloaded on 03/01/2011 (mm/dd/yyyy) from http://www.geneontology.org/GO.downloads.ontology.shtml. We downloaded GOA on 03/8/2011 from *ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/HUMAN/gene_association. goa_human.gz*. For our simulation, we used the human species of GOA, and this version of GOA has 18,141 genes. For GO, we restricted our analysis to the biological process ontology. We also excluded IEA annotations, which are the only automatically inferred annotations. We made these choices with regards to GO in order to be consistent with [38]. The seed dataset for generating gene expression data is GDS2771 from the Gene Expression Omnibus (GEO).[2] This dataset was collected by Spira et al. and it was derived from large airway epithelial cells from smokers with cancer and smokers without cancer using Affymetrix HG-U133A microarrays [49]. Spira et al. used the Robust Multi-array Average (RMA) algorithm [50] to obtain probe-level data. There are 192 samples in this dataset; there are 90 samples without cancer and 102 with cancer. The primary reason for selecting this dataset is its size. After we discuss our data generation process, we introduce a slightly modified version of SoFoCles, which is a feature selection technique that uses GO to select genes. After these important topics are developed, we explain our overall experimental methodology. We close this section with a measure that quantifies the notion of connectedness for a biological situation, or biological condition, generated by our simulation.

3.1. Simulation methods

We restrict our analysis to cases where there are only two groups, a control group and an experimental group. We are using these terms quite loosely; because our simulation should be apply to any two-class gene expression data problem. In this section, there are two main points that we discuss: (1) Algorithm 1, which is a way to define a group of differentiating genes between the control and experimental class from GO; and (2) a method for generating gene expression data using this group of genes.

Algorithm 1.

```
% α is the minimum number of genes to be significant
% β is the information content threshold
% G_enes is a list of n gene symbols
% g_i is the gene symbol at index i
% GO is the list of all GO terms
% Annotated = {(g, t) | g ∈ G_enes ∧ t ∈ GO ∧ t is annotated to g in GOA}
% Δ is the output list of gene that differentiate the control and experimental classes
Δ ← ∅
while |Δ| ⩽ α do
  i ← randomInteger(0, n − 1)
  Δ ← Δ ∪ g_i ∈ G_enes
  GOIDs ← {t | (g_i, t) ∈ Annotated ∧ IC_norm(t) ⩾ β}
  for all t ∈ GOIDs do
    Δ ← Δ ∪ {g | (g, t) ∈ Annotated}
  end for
end while
return Δ
```

[1] http://www.mathworks.com/products/matlab/.

[2] http://www.ncbi.nlm.nih.gov/geo/.

The output of Algorithm 1, Δ, is a set of genes that are differentially expressed between the control and experimental classes. This set Δ is the input to the data generation step, which we discuss later. We now provide the reader with some intuition on how this algorithm functions. Algorithm 1 starts by selecting a random gene g_i from G_{enes}, the list of all gene symbols. This gene is then appended to Δ. All the GO terms that are annotated to g_i with information content at least β are inserted in the set *GOIDs*. Next, for each GO term t in the set *GOIDs*, we find all genes annotated to t, and append all these genes to Δ. This process repeats while the size of Δ is less than or equal to the parameter α. Now that we have defined Δ, we can discuss how to generate gene expression data using this set.

3.1.1. Gene expression data generation

Our approach to gene expression data generation is based on a real gene expression simulation created by Singhal et al. [51]. This method takes a set of real gene expression profiles as a seed and then adds three levels of noise to create new gene expression profiles. In our study, we compare a control class and an experimental class. Our seed control data comes from non-cancer samples of the GEO dataset GDS2771. While this dataset has two classes, we only use the non-cancer samples because we must control the experimental class's characteristics to effectively study GO-based feature selection. Two parameters, Δ and δ, characterize the experimental class. The first parameter Δ defines the genes that are truly differentially expressed between the control and experimental classes. The second parameter δ defines the mean separation in actual gene expression between the control and experimental classes for all genes from Δ. Since GDS2771 was preprocessed with RMA, δ corresponds to units in the RMA preprocessed expression space. All other genes have a small random increase or decrease in expression level between experimental and control class's seed data. This random change in expression for each gene is modeled by normal random variable with a mean of zero and a standard deviation of 0.1. Hence for a particular Δ and δ, we create the experimental seed data from the control seed data.

To generate new synthetic gene expression data, we add three sources of noise to seed data. The three sources of noise, defined by Singhal et al., are the following: (1) systematic technical variability or inter-array variability; (2) random technical variability or intra-array variability; and (3) biological variability [51]. The

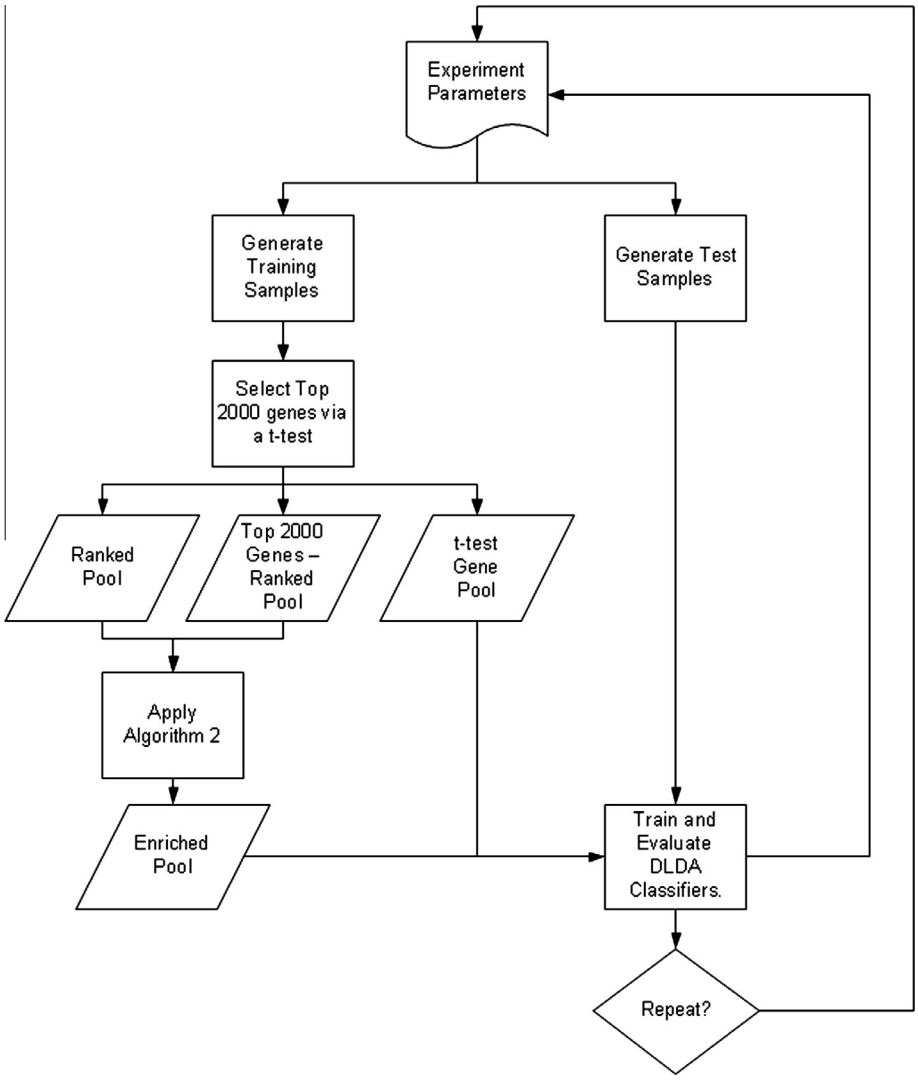

Fig. 1. An overview of an experimental run. Given Δ, δ and the number of training samples per class parameters, we generate training samples and test samples. A two-sample t-test is used rank the genes, where genes with a rank greater than 2000 are removed. Genes with the highest r absolute t-test values are used as the *rankedPool*, genes with the highest $2r$ absolute t-test values are used for the *ttestPool*. The *enrichedPool* is formed from the ranked pool using Algorithm 2. Two DLDA classifiers are trained one using the *enrichedPool* and one using the *ttestPool*. These classifiers are evaluated on the test samples. This process is repeated 20 times for a given set of parameters.

mathematical details of this method and further discussion of the three types of noise are in Appendix A.

3.2. Simulation implementation

We now discuss and justify some of our parameter settings used for our simulation and explain our experimental approach. In our experiments, we compare the accuracy of two Diagonal Linear Discriminate Analysis (DLDA) classifiers trained on data generated from two classes. One classifier is trained using a subset of genes that are found to be important from a statistical standpoint. The second classifier is trained using a subset of genes that contains half statistically important genes and half semantically important genes. The first classifier we refer to as the statistical classifier, and the second classifier we refer to as the enriched classifier. The number of genes used for training both classifiers is identical. Taking this approach, we are testing whether or not a pool of genes containing half statistically important genes and half semantically important genes is more effective than using statistically important genes only.

The central question of this study is: When is it advantageous to use GO-based feature selection? To assist in answering this question, we defined a measure called Biological Condition Annotation Level ($BCAL(G)$). We introduce $BCAL(G)$ in depth in Section 3.3. But, at this point it is enough to understand the basic idea of what $BCAL(G)$ represents. $BCAL(G)$ is calculated from a set of genes Δ and it takes on a value between zero and one. The closer $BCAL(G)$ is to one, the more connected the genes in Δ are in GO. This implies, that the closer $BCAL(G)$ is to one, the more improvement we expect from GO-based feature selection. To see how $BCAL(G)$ affects GO-based feature selection, we created ten different biological conditions with different $BCAL(G)$ values. Recall that Δ defines the set of genes differentially expressed between a control and an experimental class. Every Δ has a corresponding $BCAL(G)$ associated with it. To create an initial Δ, we applied Algorithm 1 with $\alpha = 30$, and $\beta = 1$. These parameters resulted in a Δ with $|\Delta| = 36$, where $|\Delta|$ refers to the number of genes in Δ.

We created ten new biological conditions by modifying this initial Δ, which we refer to as Δ_0. To modify Δ_0, we either added or removed genes. We added genes to Δ_0 to reduce its corresponding $BCAL(G)$ value. We removed genes to increase its $BCAL(G)$ value. Our methodology for adding genes to Δ_0 was as follows: (1) sort gene symbols lexicographically; and (2) add genes with lowest rank to Δ_0 until we achieved our target $BCAL(G)$. In many cases, we had to remove some genes from Δ_0 to obtain our target $BCAL(G)$. We refer to these ten biological conditions as $\{\Delta_1, \Delta_2, \ldots, \Delta_{10}\}$. These biological conditions have the following corresponding $BCAL(G)$ values {1.000, 0.892, 0.785, 0.696, 0.584, 0.488, 0.389, 0.291, 0.182, 0.086}. Although it is difficult to create a biological condition with an exact $BCAL(G)$, our goal was to start with $BCAL(G) = 1.000$ and reduce the $BCAL(G)$ of each successive biological condition by 0.10. The number of genes in each biological condition is {23, 26, 31, 35, 36, 39, 36, 38, 38, 37} respectively. For each biological condition Δ_i for $i \in \{1, 2, \ldots, 10\}$, we varied two parameters δ, and the number of training samples per class. The values of δ that we investigate are {0.3, 0.5, 0.7, 0.9 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5}. The number of training samples per class we explore are {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}. Since there were two classes the total number of training samples varied from 20 to 200. For each Δ_i, we performed 120 experiments; this values comes from the fact that there are 12 values of δ and 10 different training sample sizes. Each experiment was repeated 20 times, and we computed the average of the 20 replications to get the average difference in classification accuracy between the enriched classifier and the statistical classifier. We refer to the classifier using GO as enriched because the statistical genes are semantically enriched by GO. For all experiments, we fixed the number of selected genes to 40.

Another important issue regarding $BCAL(G)$ is: How does the number of genes in Δ affect GO-based feature selection? To answer this question, we focused on cases where $BCAL(G) = 1.000$, because we could efficiently control the number of genes. We then created ten new biological conditions: $\{\Delta_{11}, \Delta_{12}, \Delta_{13}, \Delta_{14}, \Delta_{15}, \Delta_{16}, \Delta_{17}, \Delta_{18}, \Delta_{19}, \Delta_{20}\}$. The number of genes in each of these biological conditions is {5, 10, 50, 100, 150, 200, 250, 300, 350, 400} respectively. The same values for δ and the number of training samples per class as above are used in the experiments.

Fig. 1 displays the steps of an experimental run. For a particular value of δ, we generate 2000 control group and 2000 experimental group training samples. A similar procedure is also applied to generate the test set. Assuming all the other parameters are fixed, we then generate a training sample permutation with the number of samples specified by the training sample size per class parameter. For example, if the training sample size per class parameter is 20, then a sample of 40 gene expression profiles would be created with 20 profiles from the control group and 20 profiles from experimental group.

To rank the genes, we applied a two-sample t-test to each gene. In order to reduce the search space, only 2000 genes with the highest absolute value t-test were kept, while all the other genes were filtered out.

After the top 2000 genes were selected, we create three pools of genes: *rankedPool*; *ttestPool*; and *enrichedPool*. The *rankedPool* contains the top r as ranked by the absolute value of the t-test. The *ttestPool* contains the top $2r$ genes as ranked by the absolute value of the t-test. For all our experiments $r = 20$. The *enrichedPool* is constructed using Algorithm 2.

Algorithm 2. Enrichment Algorithm

```
% rankedPool the top r t-test genes
% otherPool = top2000ttest − rankedPool
% similarity[i] = 0, ∀i ∈ otherPool
for all g ∈ rankedPool do
  for all g′ ∈ otherPool do
    sim_o = Sim_MAX(g, g′)
    if similarity[g′] < sim_o then
      similarity[g′] = sim_o
    end if
  end for
end for
% sort otherPool in descending order by similarity and then in
  descending order by the absolute value of the t-test.
simPool = sort(otherPool, similarity)
enrichedPool = rankedPool ∪ {g | g's index in simPool ⩽ r}
return enrichedPool
```

Algorithm 2 finds the r most semantically similar genes to the genes in the *rankedPool*. Algorithm 2 is based on the enrichment process presented in [38]. We differ from the original implementation by incorporating the absolute t-test value into the sorting activity.

This is important when using Sim_{MAX}, because many gene pairs are given the same semantic similarity. For example, any pair of genes that have a common leaf term annotation will have identical semantic similarity.

Sim_{MAX} was used because it has been shown to perform well with gene expression data [21]. In addition, in the study by Papachristoudis et al. [38] it performed consistently, and its implementation is simple.

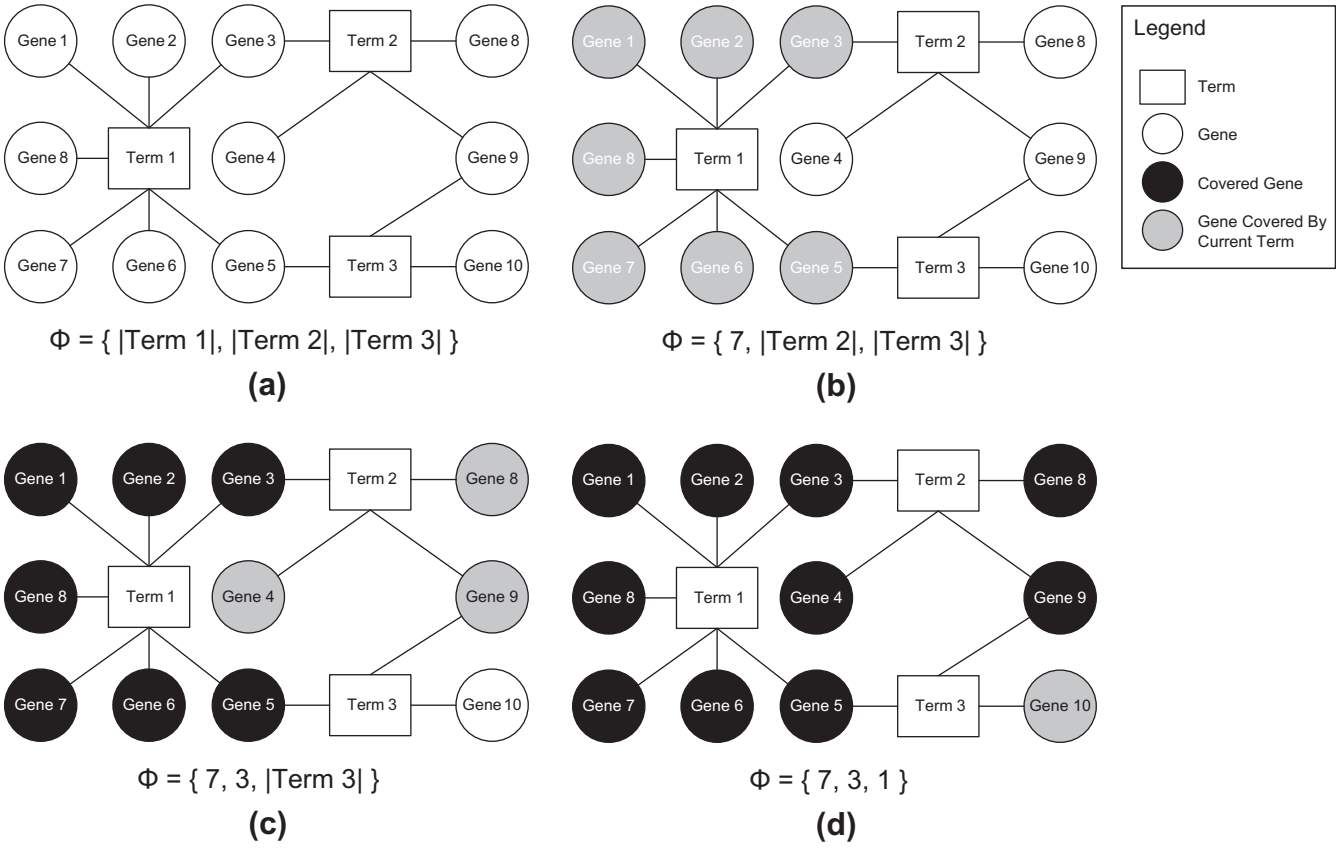

Fig. 2. Example of $BCAL(G)$ Calculation. (a) Graph G of differentially expressed genes represented by the set Δ, where $|Term\ 1|$, $|Term\ 2|$ and $|Term\ 3|$ refer to the number of uncovered genes for each term. The initial value of ϕ is $\{|Term\ 1|, |Term\ 2|, |Term\ 3|\}$. (b) Calculation of ϕ for the term with the largest number of genes associated to it (Term 1). All seven genes covered by Term 1 are marked at this point with the value of $\phi = \{7, |Term\ 2|, |Term\ 3|\}$. (c) Calculation of ϕ for the term with the most remaining unmarked genes associated with it (Term 2). Three additional genes are covered and marked with the value of $\phi = \{7, 3, |Term\ 3|\}$. (d) Calculation of ϕ for the only remaining term (Term 3). At this point the calculation of ϕ is complete with its value being $\phi = \{7, 3, 1\}$. At this stage, $BCAL(G)$ is calculated using Δ and ϕ as $BCAL(G) = \frac{\|\phi\|}{|\Delta|} = \frac{\sqrt{7^2+3^2+1^2}}{10} = 0.768$.

After the creation of the *enrichedPool*, we train two DLDA classifiers. One classifier is trained using the features in the *enrichedPool* and the other classifier is trained using the features from the *ttestPool*. Both pools contain $2r$ genes. The classifiers are evaluated using 4000 test samples that we generated earlier. This process is repeated 20 times for a given set of parameters. At the end of the experiment, the average classification accuracies, the difference in classification accuracy and the number of true positive differentially expressed genes are calculated for each pool. Finally, after all parameter values have been examined, we alter Δ to investigate other biological conditions with different annotation levels.

3.3. Biological condition annotation level

Up to this point we have not mentioned a method in detail to quantify the connectedness of a biological condition represented by Δ. To do this we must take into account how Algorithm 2 functions. Specifically, when would we expect Algorithm 2 to perform best? Suppose a biological condition is represented by all genes connected to the same GO term. Assuming this is a leaf term, the semantic similarity will be one between all pairs of genes. If this is the case, and if one of these genes is a member of the *rankedPool*, then Algorithm 2 can infer all other genes from this single gene. This base case gives us some intuition for how to define a measure that quantifies how effective Algorithm 2 will be for a given Δ.

Since the connectedness of the genes in Δ is largely dependent on GOA annotations, we will refer to the connectedness of Δ as the biological condition annotation level (*BCAL*). To discover a suitable metric that quantifies the annotation level of biological condition, one can investigate its graphical structure.

Let $G = (V, E)$, where $V = Terms \cup \Delta$ and $E = \{(g, t) \mid g \in \Delta$ is annotated by term $t \in Terms\}$ and $Terms = \{t \mid t \in GO \wedge IC_{norm}(t) \geqslant \beta' \wedge t$ is annotated to at least two genes $\in \Delta\}$. *Terms* represents the set of all GO terms annotated to at least two genes in Δ and meeting the minimum information content threshold β'. A term must be annotated to at least two genes because if it were only annotated to one gene it would contribute nothing toward discovering other genes. E is the set of all edges between genes in Δ and GO terms from the set *Terms*. The graph G is bipartite by construction, since it can be partitioned into two sets of nodes, *Terms* and Δ, where there are only edges between Δ and *Terms*. An important substructure of G, that should quantify the suitability of a biological condition for enrichment, is how well high information content terms cover the genes of G. By cover, we mean there is an edge between a term and a gene. A term that covers many genes has high degree. A biological condition that requires fewer high information content terms to cover its genes should have a higher annotation level than a biological condition that requires more terms to cover its genes. As mentioned earlier, a biological condition where all genes are covered by a single leaf term represents an ideal condition. In this case, only one significant gene is required to infer all significant genes. We now define the $BCAL(G)$ measure, which describes the set covering level of high information content terms for a particular biological condition as: $BCAL(G) = \frac{\|\phi\|}{|\Delta|}$, where G is a graph representing Δ, $\sum_{i=1}^{|Terms|} \phi_i \leqslant |\Delta|$ and $\phi_i \in \mathbb{N}$ is defined in Algorithm 3.

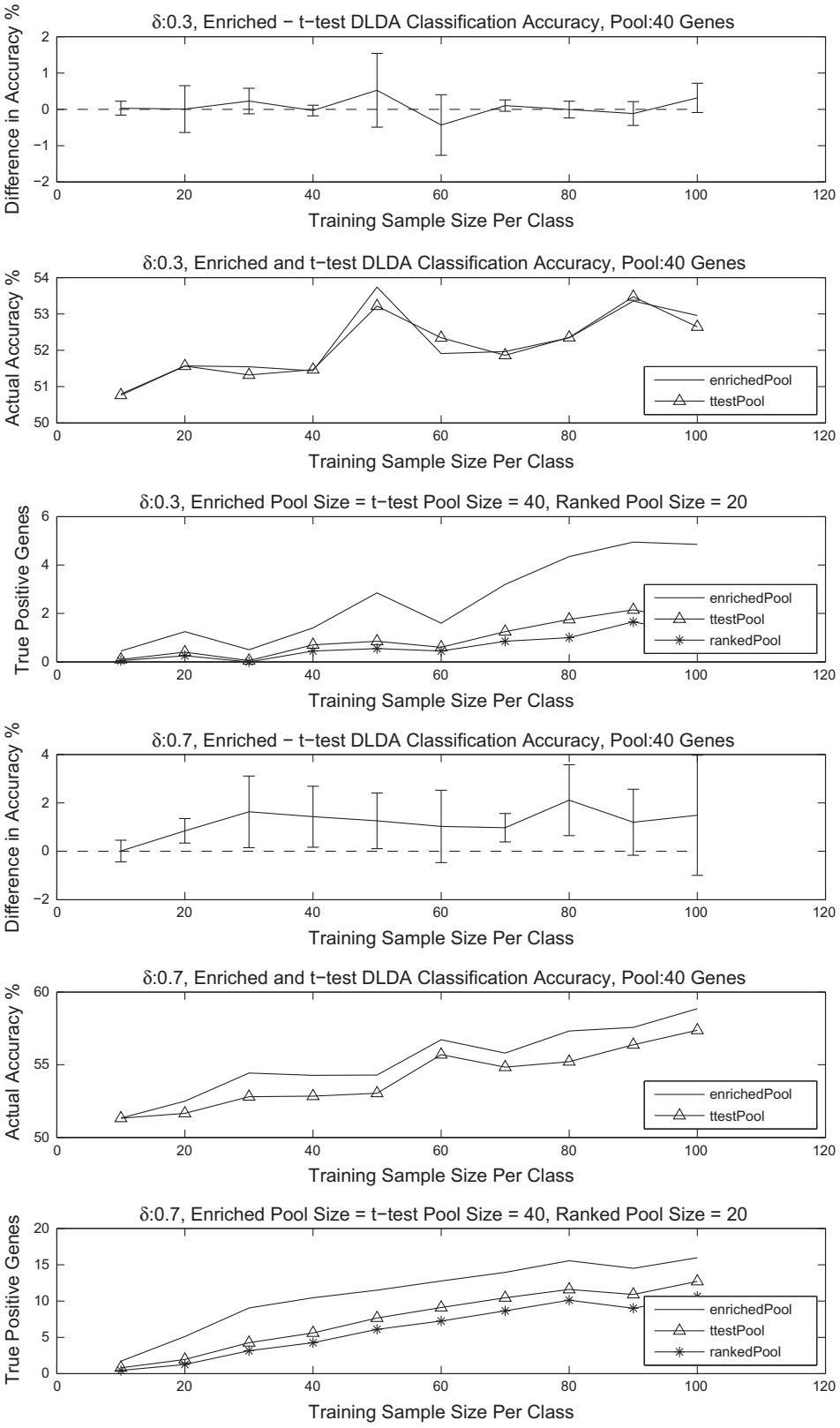

Fig. 3. A subset of simulation results for Δ_1. The results are shown for the cases where $\delta = \{0.3, 0.7\}$ and the pool size was fixed at 40. There is one plot for each value of δ. Within each of the two subplots above, there are three panels. The top panel has a 95% confidence interval of the difference in accuracy between the *enrichedPool* classifier and the *ttestPool* classifier for a given training sample size per class. The middle panel shows the average classification accuracy for each classifier by each training sample size per class. The bottom panel shows the number of true positive genes for each gene pool varying by the training size per class parameter.

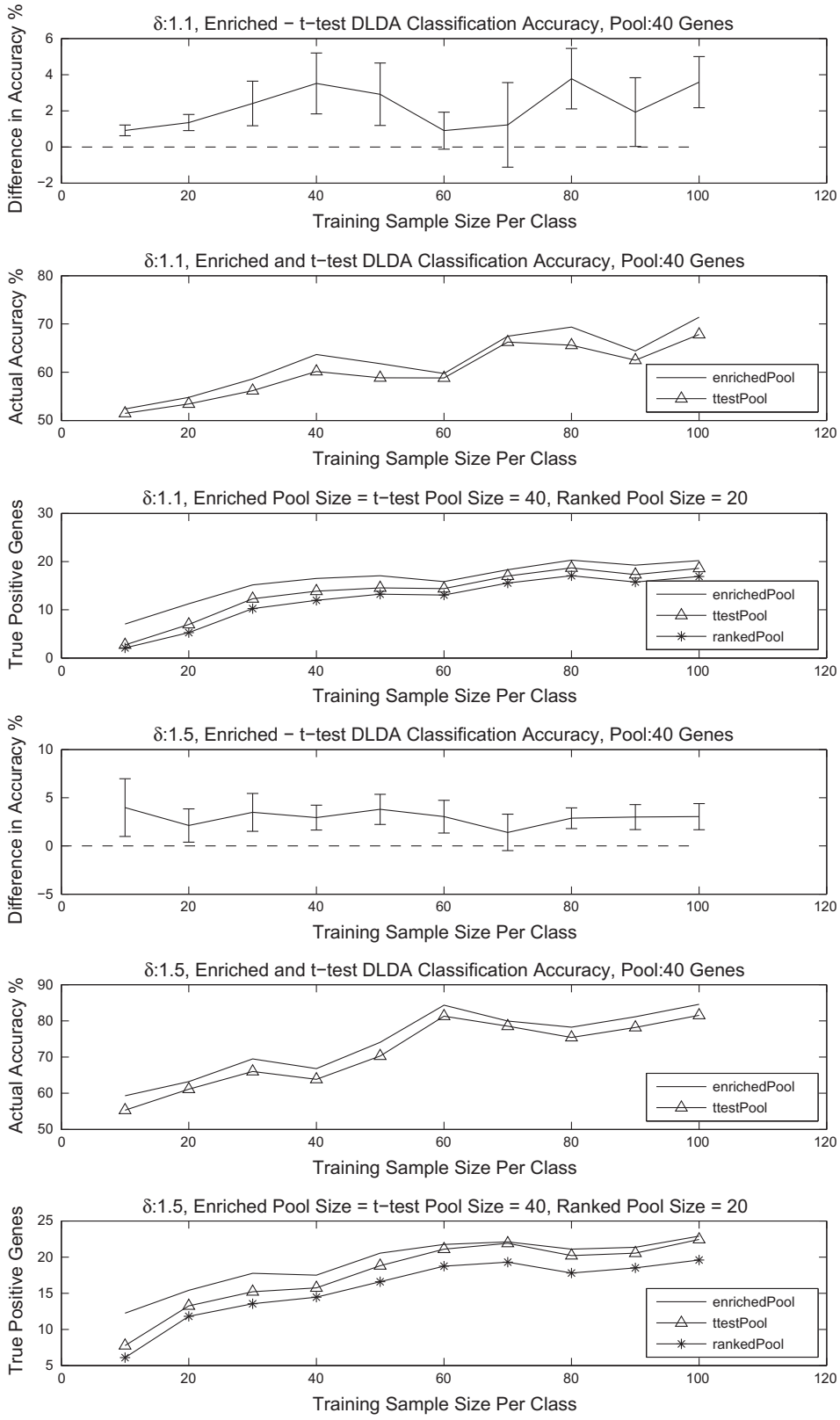

Fig. 4. A subset of simulation results for Δ_1. The results are shown for the cases where $\delta = \{0.9, 1.5\}$ and the pool size was fixed at 40. There is one plot for each value of δ. Within each of the two subplots above, there are three panels. The top panel has a 95% confidence interval of the difference in accuracy between the *enrichedPool* classifier and the *ttestPool* classifier for a given training sample size per class. The middle panel shows the average classification accuracy for each classifier by each training sample size per class. The bottom panel shows the number of true positive genes for each gene pool varying by the training size per class parameter.

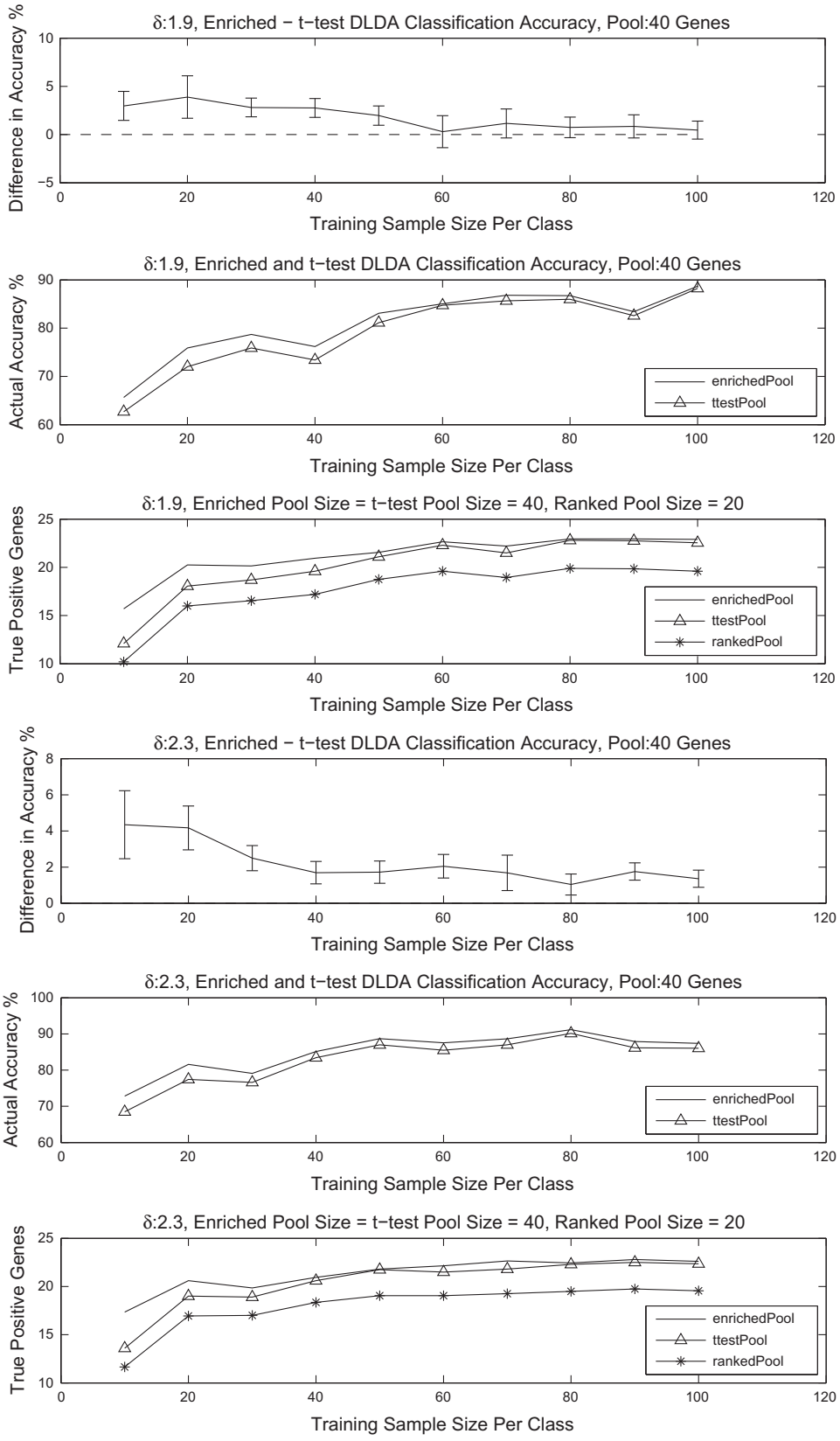

Fig. 5. A subset of simulation results for Δ_1. The results are shown for the cases where $\delta = \{1.9, 2.3\}$ and the pool size was fixed at 40. There is one plot for each value of δ. Within each of the two subplots above, there are three panels. The top panel has a 95% confidence interval of the difference in accuracy between the *enrichedPool* classifier and the *ttestPool* classifier for a given training sample size per class. The middle panel shows the average classification accuracy for each classifier by each training sample size per class. The bottom panel shows the number of true positive genes for each gene pool varying by the training size per class parameter.

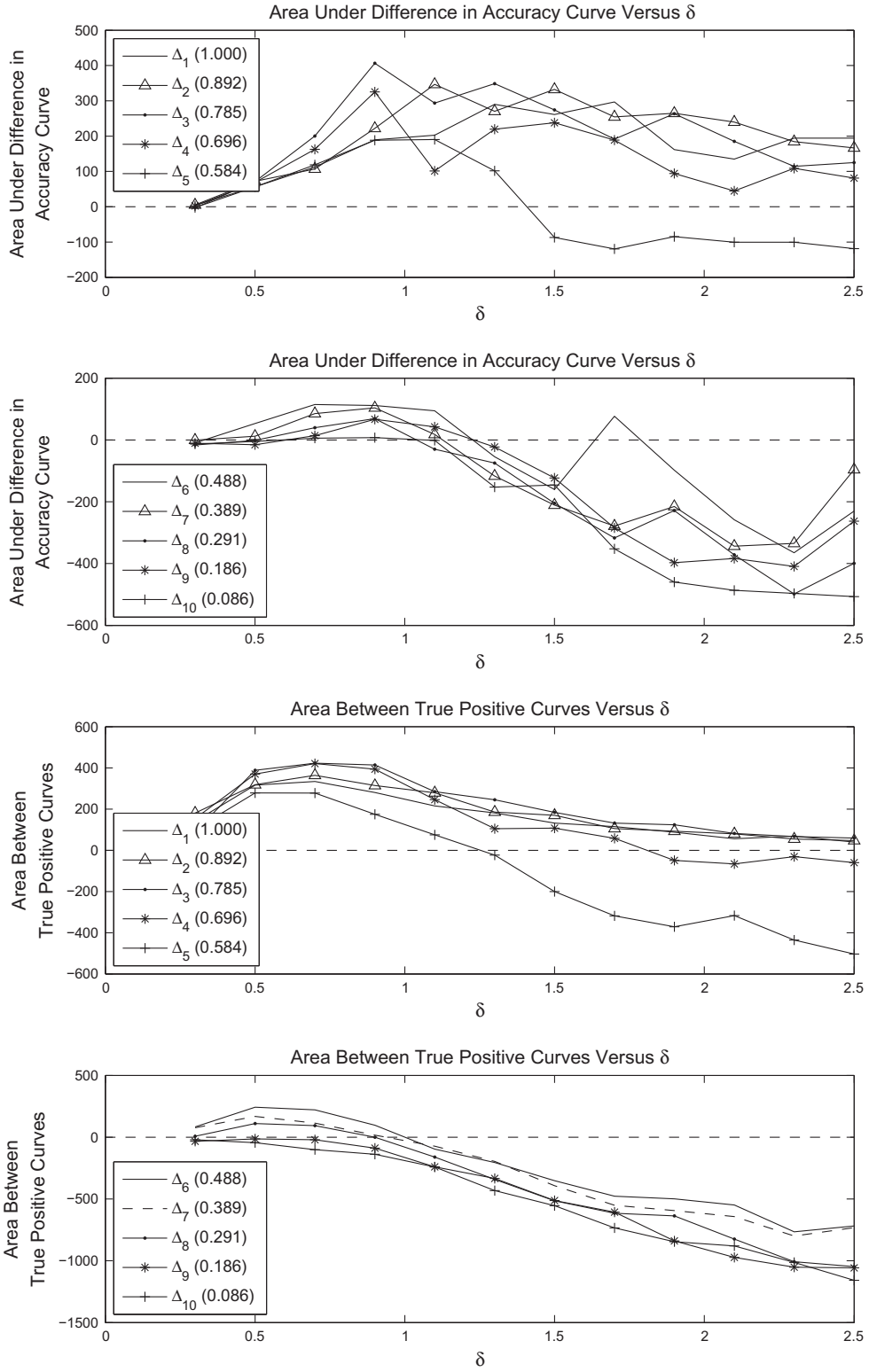

Fig. 6. A Comparison of GO-based feature selection while varying *BCAL*(*G*). In the legend, the value in parentheses next to Δ represents its corresponding *BCAL*(*G*) value. The top two plots shows the area under the difference in accuracy curve versus δ. If the curve is above zero, then there is a net improvement in classification accuracy from GO-based feature selection. When the curve is below zero, there is a reduction in classification accuracy. The bottom two plots shows the area between the true positive curves for the *enrichedPool* and the *ttestPool*. If the curve is above zero, then the *enrichedPool* has more true positives. Otherwise, the *ttestPool* had more true positives.

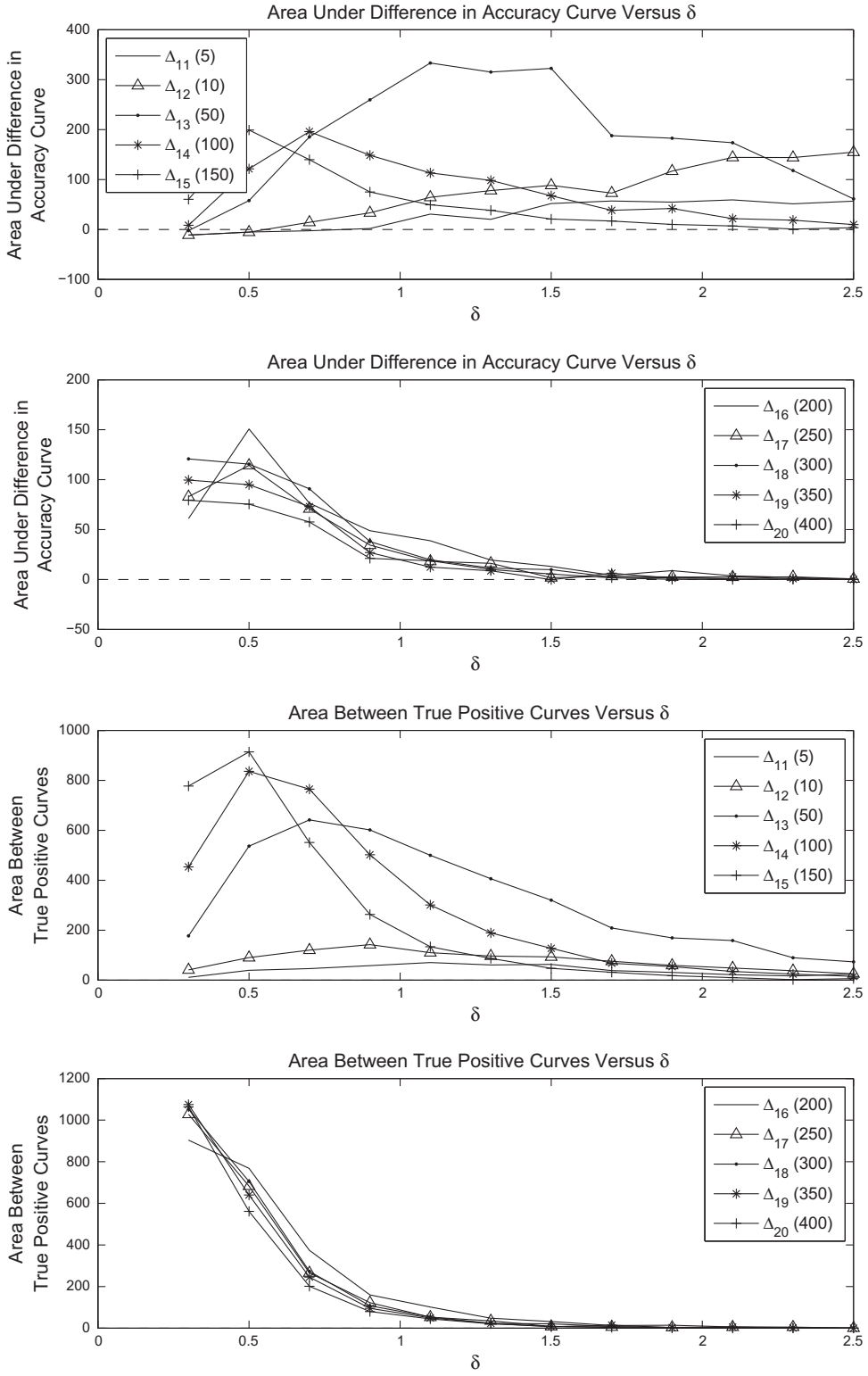

Fig. 7. A comparison of GO-based feature selection while keeping the $BCAL(G)$ fixed at one and varying the number of genes in the biological condition. In the legend of each plot, the number of genes is in parentheses next to its corresponding Δ. The top two plots show the area under the difference in accuracy curve versus δ. If the curve is above zero, then there is a net improvement in classification accuracy from GO-based feature selection. When the curve is below zero, there is a reduction in classification accuracy. The bottom two plots shows the area between the true positive curves for the *enrichedPool* and the *ttestPool*. If the curve is above zero, then the *enrichedPool* has more true positives. Otherwise, the *ttestPool* had more true positives.

Algorithm 3. BCAL Algorithm

```
% geneCoveredBy(t) = {g |(g, t) ∈ E}
% Isolated = {g | g ∈ Δ ∧ g is not annotated to any t ∈ Terms}
∀i, φ_i = 0
covered = ∅
while |covered| < |Δ| − |Isolated| do
  t ← argmax_t(|genesCoveredBy(t) − covered|)
  φ_t = |genesCoveredBy(t) − covered|
  covered = covered ∪ genesCoveredBy(t)
end while
return ||φ|| / |Δ|
```

Algorithm 3 is a greedy algorithm which approximates the minimum set cover of the terms of G, inspired by [52]. Its general concept is depicted pictorially in Fig. 2. This algorithm first finds the term t that covers the most genes and assigns ϕ_t to be the number of genes covered by term t. On the next iteration, the algorithm finds the term t' that covers the most genes that are not already covered, and it assigns the value $\phi_{t'}$ to be the number of genes covered by t', which are not already covered. This process continues until all genes are covered or only isolated genes remain. By finding the approximate set cover, we do not double count genes. This allows us to normalize the vector ϕ by $|\Delta|$. The optimal value, $BCAL(G) = 1$, occurs when all genes are annotated by a single term, which is the star graph. So all genes, g_1 and g_2, have $Sim_{MAX}(g_1, g_2) = 1$. Thus knowing one significant gene allows Algorithm 2 to infer all other significant genes through the common term. $BCAL(G)$ is normalized by dividing by $|\Delta|$ so that $0 \leqslant BCAL(G) \leqslant 1$. The reason for this is $\|\phi\| = \sqrt{\sum_{i=1}^{|Terms|} \phi_i^2} \leqslant \sqrt{\left(\sum_{i=1}^{|Terms|} \phi_i\right)^2} = \sum_{i=1}^{|Terms|} \phi_i \leqslant |\Delta|$. Dividing both sides of the inequality by $|\Delta|$ implies $\frac{\|\phi\|}{|\Delta|} \leqslant 1$. One disadvantage of $BCAL(G)$ is that it has the information content parameter β'. In our investigation, we fixed $\beta' = 1$. If $\beta' = 1$, then some biological conditions may get an artificially low $BCAL(G)$. This could occur when the genes that characterize a biological condition are related mainly through a term t with $IC_{norm}(t) < 1$. However, the β' parameter should be as high as possible, because the semantic similarity between two terms is bounded by the information content of their lowest common ancestor. So using a low β may yield an artificially high $BCAL(G)$.

We now expand on the biological meaning of $BCAL(G)$. Suppose we are comparing two groups of patients, and there are 40 true differentially expressed genes between these groups of patients. If all of these genes are annotated to a single GO leaf term, then this biological condition would have a $BCAL(G) = 1$. In this case, all the genes have gene products involved in the same specific biological process. There could be other biological processes that have been altered, but there is a single common biological process that is annotated to all 40 of the differentially expressed genes. On the other end of the spectrum, if the 40 genes either were not annotated to any leaf GO terms or each gene was only annotated to a single leaf GO term, then this condition would have a $BCAL(G) = 0$. In this case, there are biological processes that have been altered, but there is no commonality of the genes in terms of GO leaf annotations for these biological processes. That is each altered existing biological process in GO is annotated to at most one gene. If the biological condition had a $BCAL(G) = 0.5$, this could occur by having four altered biological processes each annotated to ten of the differentially expressed genes. In this case, there are at least four specific biological processes that have been altered, and each one of these biological processes has 10 unique differentially expressed genes. In essence, $BCAL(G)$ is a measure of how closely related the genes are in terms of specific biological processes. $BCAL(G)$ will be closer to one if the differentially expressed genes are involved with fewer specific biological processes. $BCAL(G)$ will be closer to zero if the differentially expressed genes have little commonality among the altered biological processes. One possible cause for low $BCAL(G)$ could be not enough information in GO for the biological condition under study.

4. Results and discussion

We investigate twenty synthetic biological conditions. We split the analysis into two groups of results. The first group of ten biological conditions ($\Delta_1, \Delta_2, \ldots, \Delta_{10}$) that we analyze have $BCAL(G)$ values varying from 0.086 to 1.000. The seed biological condition Δ_0 is generated using Algorithm 1, and we modify this biological condition to create the others. This biological condition Δ_0 has $|\Delta_0| = 36$ with $BCAL(G) = 0.68$.

The second group of ten biological conditions ($\Delta_{11}, \Delta_{12}, \ldots, \Delta_{20}$) all have $BCAL(G) = 1.000$. However, these biological conditions have differing cardinalities from five to 400. The purpose of analyzing this group of biological conditions is to discover how the number of genes affects the improvement of GO-based feature selection. In this section, we first discuss the results of one biological condition, Δ_1, in detail. Next, we discuss the summary of the results for $\Delta_1, \Delta_2, \ldots, \Delta_{10}$. We then discuss the summary of the results for $\Delta_{11}, \Delta_{12}, \ldots, \Delta_{20}$.

4.1. Biological condition Δ_1

We now analyze the results for Δ_1. To recap, this condition has a $BCAL(G) = 1.00$, and $|\Delta_1| = 23$. Fig. 3 displays the results of our experiments when $\delta = 0.3$ and 0.7. Fig. 4 presents the results when $\delta = 1.1$ and 1.5. Fig. 5 shows the results when $\delta = 1.9$ and 2.3. Each figure has two subplots. Each subplot corresponds to a particular value of δ, and there are three panels within each subplot. The top panel displays a 95% confidence interval for the difference in classification accuracy between the classifier trained using the *enrichedPool* of genes and the classifier trained using the *ttestPool* of genes. The middle panel shows the average classification accuracy for each classifier over 20 experimental repetitions. Finally, the bottom panel shows the number of true positive genes for each gene pool.

When $\delta = 0.3$, there is no improvement in classification accuracy when using the *enrichedPool*. There is an improvement in the detection of true positive genes for the GO-based feature selection. For example, when the number of training samples per class is around 100, the *enrichedPool* has an average of about five true positives where as the *ttestPool* has an average of two. There was no improvement in classification accuracy because a $\delta = 0.3$ is insufficient in differentiating between the classes with so few genes.

In the case where $\delta = 0.7$, there is an improvement in classification accuracy when the number of training samples is greater than 10 per class. However, this improvement is quite small, averaging around 2% for most sample sizes. The overall classification accuracy is between 50–60%, which is quite low. The *enrichedPool* contained more true positives than the *ttestPool* for all training sample sizes.

As δ increases to 1.1, the amount of improvement is larger. The average improvement approaches 4% when the number of training samples is 40 per class, 80 per class and 100 per class. A similar pattern occurs when $\delta = 1.5$. When δ is increased to 1.9 the improvement is close to 5% at small sample sizes and approaches zero as the training sample size increases. The situation is similar for the case where $\delta = 2.3$. The improvement decreases toward zero

because both the *enrichedPool* and the *ttestPool* on average contain nearly all 23 of the genes from Δ_1.

Inspecting the full range of results reveals a pattern. Essentially, the results follow a roughly concave down curve, which shifts from large training sample sizes to smaller sample sizes as δ increases. These results indicate that there is an increase in classification accuracy in almost all cases when $BCAL(G) = 1.000$. But, this is the ideal case, what happens when we decrease $BCAL(G)$?

4.2. Results for different BCAL(G) values

It is clear, when $BCAL(G) = 1.000$ that it is almost always beneficial to use GO-based feature selection. However, this represents the ideal case. It is quite unlikely that real gene expression data will have $BCAL(G) = 1.000$. In this section, we compare Δ_1, Δ_2, ..., Δ_{10}. We cannot display all the results for each biological condition like we did for Δ_1 in the previous section, because this would require many pages. So we opted for a summarization approach, which calculates a single value for each plot. To understand this value, recall in Figs. 3–5 we are comparing three variables: δ, the training sample size, and the difference in classification accuracy. So to condense this into a two variable comparison, we calculate the area under the difference in accuracy curve over the number of training samples. In the case of Δ_1 when $\delta = 0.3$, we calculate the area under the curve in the top panel of Fig. 3. Doing this gives a single value at each value of δ for each biological condition. If the area under the curve is positive, then there is a net improvement in using GO-based feature selection. If the area under the curve is negative then GO-based feature selection does not provide a net improvement. In fact, if the area under the curve is negative, then the GO-based feature selection provides a net decrease in classification accuracy over the range of training sample sizes. The advantage of doing this summarization approach is that it allows us to see the overall trend on how $BCAL(G)$ affects the classification accuracy. This in turn provides us with a concise visualization of all the results. The downside is that we lose details on when GO-based feature selection leads to an increase in classification accuracy. For example, GO-based feature selection could increase classification accuracy at small sample sizes then decrease classification accuracy at large sample sizes. This information is lost when we calculate the area under the curve. In addition to the area under the difference in accuracy curve, we also calculate the area between the *enrichedPool* and *ttestPool* true positive curves.

Fig. 6 displays the summary of the area under the difference in accuracy curve and the area between the *enrichedPool* true positive curve and the *ttestPool* true positive curve. When $BCAL(G) \geqslant 0.696$, there is a net improvement in classification accuracy for the classifier that uses the *enrichedPool*. When $\delta = 0.3$ there is no improvement in accuracy. The GO-based feature selection provides a net improvement until $0.9 \leqslant \delta \leqslant 1.1$ where the improvement peaks. After the peak, the improvement levels off when $BCAL(G) \geqslant 0.696$. In the cases where $BCAL(G) < 0.696$, the GO-based feature selection begins to result in a reduction in classification accuracy, when $\delta \geqslant 1.5$. It is interesting to note, that both biological conditions Δ_2 and Δ_3 outperform Δ_1 even though both these biological conditions have lower $BCAL(G)$ values. We believe this is due to the fact that the pool size is 40 and the $|\Delta_1| = 23$. The $|\Delta_2| = 26$ and $|\Delta_3| = 31$, so they are closer in size to the number of selected genes. So with Δ_1, Algorithm 2 runs out of true positive genes to add to the *enrichedPool*. Another interesting observation is when $\delta > 1$, the biological condition with $BCAL(G) = 0.696$ (Δ_4) yielded more true positives than Δ_1 and Δ_2. We believe this is also caused by the cardinality of Δ_4, which is 35.

The true positive plot of Fig. 6 shows some important ideas. In the cases where $BCAL(G) \geqslant 0.584$, the overall trend for area between the true positive curves is that they start out small when $\delta = 0.3$, and they reach their peak when $\delta = 0.7$. The curves then start to approach zero as δ increases. However, for both cases where $BCAL(G) = 0.696$ and $BCAL(G) = 0.584$, the *enrichedPool* begins to contain fewer true positives than the *ttestPool*. Δ_4 falls below zero at $\delta = 1.7$, and Δ_5 falls below zero at $\delta = 1.3$. When $BCAL(G) \leqslant 0.488$ the curves reach their peaks even earlier. In the case of $BCAL(G) = 0.086$, the peak occurs at $\delta = 0.3$. In all cases where $BCAL(G) \leqslant 0.488$ the curves fall below zero before $\delta = 1$. In the worst cases, the area between the *enrichedPool* and the *ttestPool* curves is near -1000. The magnitude of this is more than twice as large as the best cases, which are near 400. This suggests GO-based feature selection can be significantly worst than its potential benefits when $BCAL(G) \leqslant 0.291$ and $\delta \geqslant 2.3$.

4.3. Results for same BCAL(G) values

One important question we have not investigated is: Given a fixed $BCAL(G)$, how does the number of genes affect the difference in classification accuracy? To answer this question, we fix $BCAL(G) = 1.000$ and vary the number of genes. We created ten biological conditions Δ_{11}, Δ_{12}, ..., Δ_{20} with the number of genes increasing from 5 to 10, then increasing from 50 to 400 by 50 genes at a time. This allows us to assess the efficacy of GO-based feature selection when the number of genes in a biological condition becomes inconvenient, that is either very large or very small. We fixed $BCAL(G) = 1$ because we can control the number genes in this case much more than we can in other cases. We also fixed the number of genes selected for classification at 40, the same as before. Fig. 7 displays the results of these simulation experiments.

There are some trends in Fig. 7 worth discussing. In the top two plots, we see the area under the difference in accuracy curves. Similar to the Fig. 6, a positive value indicates a net improvement in using the *enrichedPool* for classification as opposed to the *ttestPool*. The *enrichedPool* provides essentially no improvement for Δ_{11}. For the case of Δ_{12} the *enrichedPool* provides improvement when $\delta > 1$ and there is little to no improvement when $\delta \leqslant 1$. The biological condition Δ_{13} has a similar shape as Δ_1 from Fig. 6. When the number of genes is increased to 100 for Δ_{14} the amount of improvement significantly declines. This implies the *t*-test was able to identify many more true positives with Δ_{14} as compared to Δ_{13}. However, when $\delta < 0.7$, the *enrichedPool* yielded better performance as compared to *ttestPool* when the number of genes in the biological conditions was at least 150. If we inspect the bottom two plots of Fig. 7, we can understand this situation more clearly. When $\delta = 0.3$, as the number of genes in a biological condition increases, the *enrichedPool* contains more true positive genes. This occurs because there is a greater chance for true positive genes to be found in the initial *rankedPool* of genes. This allows Algorithm 2 to add many semantically similar genes to the *enrichedPool*. This improvement converges as the number of genes in the biological conditions approach 250. When $\delta > 1$ and the number of genes in a biological condition is greater than 150, there is little to no benefit of using GO-based feature selection. Therefore, we recommend against using GO-based feature selection for biological conditions with larger than 150 genes and $\delta > 1$. If $BCAL(G) < 1$, then we expect similar reductions in improvement.

5. Summary and conclusion

This study addresses the question of *when* to use GO-based feature selection effectively, whereas previous studies have developed methods on *how* to use GO effectively for feature selection. To investigate this question, we created a simulation. The first step of the simulation process is Algorithm 1, which outputs a set of genes. These genes are differentially expressed between a control

class and an experimental class. This set of genes represents a biological condition, and is denoted by Δ. We generate synthetic gene expression data using real data collected from large airway epithelial cells. The data from the experimental class (representing an altered biological condition) is based on the control class, except the genes that are in the set Δ have their expression either increased or decreased. The magnitude of increase or decrease for each gene in Δ is governed by the parameter δ. These two datasets constitute the seed of our data generation process, which generates new samples with additional noise. We train two DLDA classifiers on data generated from these seed datasets. One classifier uses only statistical properties to select genes, and the other classifier uses statistical properties in conjunction with semantic similarity in GO to select genes. We define a measure called $BCAL(G)$, which quantifies the annotation level, or connectedness of the genes and terms, in a biological condition.

We have five main conclusions from our simulations. First, it is beneficial to use GO-based feature selection when $BCAL(G) \geqslant 0.696$. $BCAL(G)$ could be calculated from a list of potential genes. This potential list may be found via a literature search. In practice, if a biologist expects that the differentially expressed genes are likely to be related by a small number of specific GO terms ($BCAL(G)$ closer to 1), then it is likely that using GO-based feature selection will improve classification. Second, when $BCAL(G) \leqslant 0.389$ (Δ_7–Δ_{10}), statistical feature selection outperforms GO-based feature selection except for uncommon cases presented in Fig. 6 in the second plot from the top where $0.5 \leqslant \delta \leqslant 1.2$. Practically, if a biologist expects the differentially expressed genes not to have a close relationship in GO ($BCAL(G)$ closer to 0), then it is not recommended to use GO-based feature selection. Third, when $0.389 < BCAL(G) < 0.696$, GO-based feature selection provides improvement between 0% and 9% when $\delta < 1$. However, the average improvement across all training sample sizes varies from 0% to 2%. In practice, if the potential list of genes does not fall into the two previous cases, then it may fall into this category. However, GO-based feature selection is only effective in this case if the average change in expression (δ) is less than one unit in the RMA-normalized space. Fourth, If $BCAL(G) = 1$ is fixed and we increase the number of genes in Δ beyond 50 with $\delta \geqslant 0.7$, then the improvement from GO-based feature selection decreases. Here we mean the average improvement starts off larger and gets smaller as the number of important genes increases. This is assuming the feature selection pool size is fixed. In particular, when the number of genes in a biological condition is greater than 150, and $\delta > 1$, we do not recommend using GO-based feature selection. Practically, a biologist should not use GO to improve feature selection if the number of potential genes is more than 150. Fifth, we do not recommend using GO-based feature selection if the number of genes in a biological condition is less than 10.

While our simulation provides an understanding of GO-based feature selection, it has a few limitations. Our synthetic data was generated from real data that was preprocessed using RMA. Hence we believe our conclusions are valid for datasets preprocessed using RMA. Other than RMA, the values of δ may vary for different preprocessing methods. Other limitations of our study stem from the fact that we restrict our analysis to a single semantic similarity measure, feature selection method, and classification method. However, it is possible that GO-based feature selection would yield improvement on similar $BCAL(G)$ ranges for different semantic similarity measures, feature selection, and classification methods. Our future research will address these limitations.

Overall our simulation provides insight into GO-based feature selection. Specifically, our simulation shows *when* it is beneficial to use SoFoCles-like feature selection. We believe our simulation can help researchers develop classifiers that utilize GO-based feature selection more effectively. In addition, our simulation may positively impact enrichment analysis tools and functional annotation-based analysis, because it could allow researchers to specify biological conditions from GO and compare the effectiveness of these algorithms in detecting the specified biological condition. This comparison may have a positive impact on projects like the Connectivity Map [53] because it could lead to improved annotation-level representations of biological conditions. Furthermore, $BCAL(G)$ may improve single-gene enrichment analysis tools because it could be used to find groups of terms that best cover a relevant gene list at differing information content thresholds.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.jbi.2013.07.008.

References

[1] van't Veer LJ, Bernards R. Enabling personalized cancer medicine through analysis of gene-expression patterns. Nature 2008;452(7187):564–70.
[2] Dudoit S, Fridlyand J. Introduction to classification in microarray experiments. In: Berrar DP, Granzow M, Dubitzky W, editors. A practical approach to microarray data analysis. Springer; 2009. p. 132–49.
[3] Oshlack A, Robinson M, Young M. From rna-seq reads to differential expression results. Genome Biol 2010;11(12):220.
[4] Dubitzky W, Granzow M, Downes CS, Berrar DP. Introduction to microarray data analysis. In: Berrar DP, Granzow M, Dubitzky W, editors. A practical approach to microarray data analysis. Springer; 2009. p. 1–46.
[5] Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classification of tumors using gene expression data. J Am Stat Assoc 2002;97(457):77–87.
[6] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 1999;286(5439):531–7.
[7] Li L, Weinberg CR, Darden TA, Pedersen LG. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 2001;17(12):1131–42.
[8] Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 2000;16(10):906–14. http://dx.doi.org/10.1093/bioinformatics/16.10.906.
[9] Leung Y, Hung Y. A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. IEEE/ACM Trans Comput Biol Bioinform 2010;7(1):108–17. http://dx.doi.org/10.1109/TCBB.2008.46.
[10] Saeys Y, Inza In, Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics (Oxford, England) 2007;23(19):2507–17.
[11] Speed TP, editor. Statistical analysis of gene expression microarray data. Chapman and Hall; 2003.
[12] Jafari P, Azuaje F. An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors. BMC Med Inform Decis Mak 2006;6(1):27.
[13] Thomas JG, Olson JM, Tapscott SJ, Zhao LP. An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. Genome Res 2001;11(7):1227–36.
[14] Bontempi G. A blocking strategy to improve gene selection for classification of gene expression data. IEEE/ACM Trans Comput Biol Bioinform 2007;4:293–300.
[15] Duan K-B, Rajapakse J, Wang H, Azuaje F. Multiple svm-rfe for gene selection in cancer classification with expression data. IEEE Trans NanoBiosci 2005;4(3):228–34. http://dx.doi.org/10.1109/TNB.2005.853657.
[16] Saraswathi S, Sundaram S, Sundararajan N, Zimmermann M, Nilsen-Hamilton M. Icga-pso-elm approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. IEEE/ACM Trans Comput Biol Bioinform 2011;8(2):452–63. http://dx.doi.org/10.1109/TCBB.2010.13.
[17] Paul TK, Iba H. Prediction of cancer class with majority voting genetic programming classifier using gene expression data. IEEE/ACM Trans Comput Biol Bioinform 2009;6(2):353–67.
[18] Ashburner M. Gene ontology: tool for the unification of biology. Nat Genet 2000;25:25–9.
[19] Barrell D, Dimmer E, Huntley RP, Binns D, O Donovan C, Apweiler R. The GOA database in 2009-an integrated gene ontology annotation resource. Nucleic Acids Res 2009;37(Suppl. 1):D396–403. http://dx.doi.org/10.1093/nar/gkn803.
[20] Lord PW, Stevens RD, Brass A, Goble CA. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. Bioinformatics 2003;19(10):1275–83.
[21] Sevilla J, Segura V, Podhorski A, Guruceaga E, Mato J, Martinez-Cruz L, et al. Correlation between gene expression and go semantic similarity. IEEE/ACM Trans Comput Biol Bioinform 2005;2(4):330–8.

[22] Huang DW, Sherman BT, Lempicki RA. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Res 2009;37(1):1–13.
[23] Khatri P, Draghici S, Ostermeier G, Krawetz SA. Profiling gene expression using onto-express. Genomics 2002;79(2):266–70.
[24] Doniger S, Salomonis N, Dahlquist K, Vranizan K, Lawlor S, Conklin B. Mappfinder: using gene ontology and genmapp to create a global gene-expression profile from microarray data. Genome Biol 2003;4(1):R7.
[25] Zeeberg B, Feng W, Wang G, Wang M, Fojo A, Sunshine M, et al. Gominer: a resource for biological interpretation of genomic and proteomic data. Genome Biol 2003;4(4):R28.
[26] Dennis G, Sherman B, Hosack D, Yang J, Gao W, Lane HC, et al. David: database for annotation, visualization, and integrated discovery. Genome Biol 2003;4(5):P3. this is the first version of this article to be made available publicly. A peer-reviewed and modified version is now available in full at <http://genomebiology.com/2003/4/9/R60>.
[27] Hosack D, Dennis G, Sherman B, Lane H, Lempicki R. Identifying biological themes within lists of genes with ease. Genome Biol 2003;4(10):R70. a previous version of this manuscript was made available before peer review at <http://genomebiology.com/2003/4/6/P4>.
[28] Castillo-Davis CI, Hartl DL. Genemerge-post-genomic analysis, data mining, and hypothesis testing. Bioinformatics 2003;19(7):891–2.
[29] Berriz GF, King OD, Bryant B, Sander C, Roth FP. Characterizing gene sets with funcassociate. Bioinformatics 2003;19(18):2502–4.
[30] Alterovitz G, Xiang M, Mohan M, Ramoni MF. Go pad: the gene ontology partition database. Nucleic Acids Res 2007;35(Suppl. 1):D322–7. http://dx.doi.org/10.1093/nar/gkl799.
[31] Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci USA 2005;102(43):15545–50. http://dx.doi.org/10.1073/pnas.0506580102.
[32] Mootha VK, Lindgren CM, Eriksson K-F, Subramanian A, Sihag S, Lehar J, et al. Pgc-1[alpha]-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. Nat Genet 2003;34(3):267–73.
[33] Watters JW, Roberts CJ. Developing gene expression signatures of pathway deregulation in tumors. Mol Cancer Ther 2006;5(10):2444–9. http://dx.doi.org/10.1158/1535-7163.mct-06-0340.
[34] Lee E, Chuang H-Y, Kim J-W, Ideker T, Lee D. Inferring pathway activity toward precise disease classification. PLoS Comput Biol 2008;4(11):e1000217. http://dx.doi.org/10.1371/journal.pcbi.1000217.
[35] Yang X, Regan K, Huang Y, Zhang Q, Li J, Seiwert TY, et al. Single sample expression-anchored mechanisms predict survival in head and neck cancer. PLoS Comput Biol 2012;8(1):e1002350. http://dx.doi.org/10.1371/journal.pcbi.1002350.
[36] Chabalier J, Mosser J, Burgun A. A transversal approach to predict gene product networks from ontology-based similarity. BMC Bioinformatics 2007;8(1):235.
[37] Chagoyen M, Carazo J, Pascual-Montano A. Assessment of protein set coherence using functional annotations. BMC Bioinformatics 2008;9(1):444.
[38] Papachristoudis G, Diplaris S, Mitkas PA. Sofocles: feature filtering for microarray classification based on gene ontology. J Biomed Inform 2010;43(1):1–14.
[39] Qi J, Tang J. Gene ontology driven feature selection from microarray gene expression data. In: 2006 IEEE symposium on computational intelligence and bioinformatics and computational biology, 2006 (CIBCB '06); 2006. p. 1–7.
[40] Qi J, Tang J. Integrating gene ontology into discriminative powers of genes for feature selection in microarray data. In: Proceedings of the 2007 ACM symposium on applied computing, SAC '07. New York, NY, USA: ACM; 2007. p. 430–4.
[41] Gillies CE, Siadat M-R, Patel NV, Wilson GD. Gene ontology based simulation for feature selection. In: International conference on knowledge discovery and information retrieval; 2011. p. 294–302.
[42] Gruber T. Ontology. In: Encyclopedia of database systems. Springer-Verlag; 2009. p. 1963–5.
[43] Robinson PN, Bauer S. Introduction to bio-ontologies. Boca Raton, Florida: CRC Press; 2011.
[44] Seco N, Veale T, Hayes J. An intrinsic information content metric for semantic similarity in wordnet. In: de Mántaras RL, Saitta L, editors. ECAI. IOS Press; 2004. p. 1089–90.
[45] Resnik P. Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the 14th international joint conference on artificial intelligence, vol. 1. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1995. p. 448–53.
[46] Jiang JJ, Conrath DW. Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of international conference research on computational linguistics; 1997.
[47] Lin D. An information-theoretic definition of similarity. In: Proceedings of the 15th international conference on machine learning; 1998. p. 296–304.
[48] Pesquita C, Faria D, Falco AO, Lord P, Couto FM. Semantic similarity in biomedical ontologies. PLoS Comput Biol 2009;5(7):e1000443.
[49] Spira A, Beane JE, Shah V, Steiling K, Liu G, Schembri F, et al. Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. Nat Med 2007;13(3):361–6.
[50] Bolstad B, Irizarry R, strand M, Speed T. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics 2003;19(2):185–93.
[51] Singhal S, Kyvernitis CG, Johnson SW, Kaiser LR, Liebman MN, Albelda SM. Microarray data simulator for improved selection of differentially expressed genes. Cancer Biol Ther 2003;2:384–92.
[52] Vazirani VV. Approximation algorithms. Germany: Springer-Verlag; 2001.
[53] Lamb J, Crawford ED, Peck D, Modell JW, Blat IC, Wrobel MJ, et al. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. Science 2006;313(5795):1929–35.