BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 17 2011, pages 2391–2398
doi:10.1093/bioinformatics/btr337

Gene expression

Advance Access publication July 9, 2011

Annotation concept synthesis and enrichment analysis: a logic-based approach to the interpretation of high-throughput experiments

Mikhail Jiline[1,*], Stan Matwin[1,2] and Marcel Turcotte[1]

[1]School of Information Technology and Engineering, University of Ottawa, 800 King Edward Avenue, Ottawa, Ontario, K1N 6N5 Canada and [2]Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Associate Editor: John Quackenbush

ABSTRACT

Motivation: Annotation Enrichment Analysis (AEA) is a widely used analytical approach to process data generated by high-throughput genomic and proteomic experiments such as gene expression microarrays. The analysis uncovers and summarizes discriminating background information (e.g. GO annotations) for sets of genes identified by experiments (e.g. a set of differentially expressed genes, a cluster). The discovered information is utilized by human experts to find biological interpretations of the experiments.

However, AEA isolates and tests for overrepresentation only individual annotation terms or groups of similar terms and is limited in its ability to uncover complex phenomena involving relationship between multiple annotation terms from various knowledge bases. Also, AEA assumes that annotations describe the whole object of interest, which makes it difficult to apply it to sets of compound objects (e.g. sets of protein–protein interactions) and to sets of objects having an internal structure (e.g. protein complexes).

Results: We propose a novel logic-based Annotation Concept Synthesis and Enrichment Analysis (ACSEA) approach. ACSEA fuses inductive logic reasoning with statistical inference to uncover more complex phenomena captured by the experiments. We evaluate our approach on large-scale datasets from several microarray experiments and on a clustered genome-wide genetic interaction network using different biological knowledge bases. The discovered interpretations have lower P-values than the interpretations found by AEA, are highly integrative in nature, and include analysis of quantitative and structured information present in the knowledge bases. The results suggest that ACSEA can boost effectiveness of the processing of high-throughput experiments.

Contact: mjiline@site.uottawa.ca

Received on April 28, 2010; revised on March 31, 2011; accepted on May 3, 2011

1 INTRODUCTION

High-throughput methods, such as expression microarrays, promoter microarrays, genome-wide physical and genomic interaction screens, while allowing to monitor the behavior of the cell as a whole, are generating wealth of information that needs to be studied and interpreted. One of the main challenges of modern bioinformatics is to develop methods and techniques that can help inferring knowledge from accumulated datasets and large-scale experimental data.

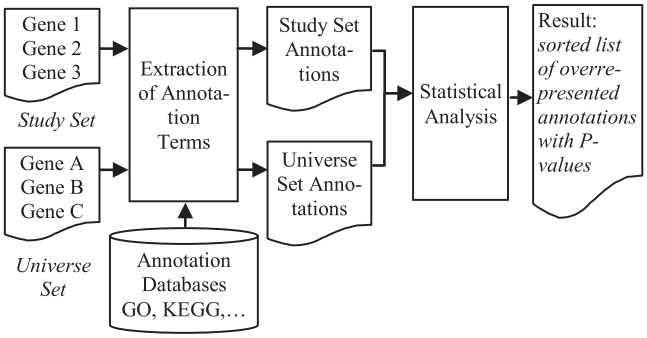

Gene 1
Gene 2
Gene 3
Study Set
Gene A
Gene B
Gene C
Universe Set
Extraction of Annotation Terms
Annotation Databases GO, KEGG,…
Study Set Annotations
Universe Set Annotations
Statistical Analysis
Result: sorted list of overrepresented annotations with P-values

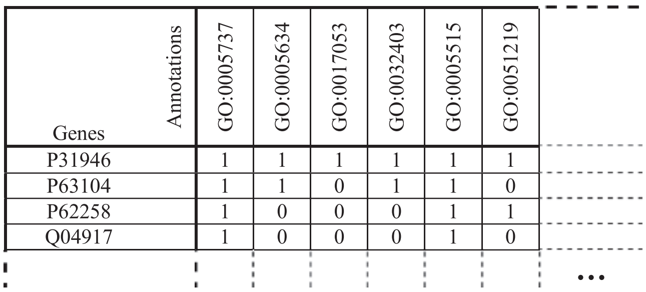
Fig. 1. AEA approach. *Study Set* is a set of genes identified by an experiment (such as differentially expressed genes or one of the clusters). *Universe Set* is the set of all the genes that participated in the experiment or some other reference set of genes that the study set will be compared against. *Annotation Database* is a source of annotations attached to genes. *Result* of the analysis is a set of annotations that are over- or underrepresented in the Study Set comparing to the Universe Set.

High-throughput experimental techniques typically generate sets of genes that require further investigation. For example, such sets may be produced by clustering microarray data or by extracting differentially expressed genes. The sets usually contain dozens or hundreds of genes, and their biological interpretation poses significant challenge to biology experts.

Recently a number of algorithms have been developed to help experts interpreting experimental data. One of the most popular approaches is Annotation Enrichment Analysis (AEA) (Huang *et al*., 2009; Khatri and Draghici, 2005). AEA uses the biological knowledge already accumulated in public databases to systematically examine large lists of genes trying to suggest biological interpretations of the experimental data. AEA algorithms extract descriptive information (called gene annotations) characterizing each gene and compare the statistical distributions of gene annotations between the gene set of interest (known as study set) and the rest of the genome or the rest of the microarray (known as universe set). Figure 1 illustrates the approach.

AEA is agnostic to the way universe and study sets are constructed. It contrasts it with other approaches,

[*]To whom correspondence should be addressed.



Genes / Annotations	GO:0005737	GO:0005634	GO:0017053	GO:0032403	GO:0005515	GO:0051219	…
P31946	1	1	1	1	1	1	
P63104	1	1	0	1	1	0	
P62258	1	0	0	0	1	1	
Q04917	1	0	0	0	1	0	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	…

Fig. 2. Bag-of-annotations data model.

such as Reconstruction of Formal Temporal Logic Models (Ramakrishnan *et al.*, 2005) which implicitly includes temportal annotations into analysis.

A variety of algorithms and tools are based on the AEA framework. They differ by the knowledge bases used as source for annotations (Gene Ontology: a controlled vocabulary of gene's attributes, KEGG: pathway information, BIND: protein–protein interactions, etc. (Alibés *et al.*, 2008; Al-Shahrour *et al.*, 2004; Minguez *et al.*, 2007; Sherman *et al.*, 2007), statistical hypothesis testing models (χ^2, Fisher's exact test, Binomial distribution, Hypergeometric distribution, etc.), types and organization of annotation terms, and sets of reference genes (Huang *et al.*, 2009; Khatri and Draghici, 2005).

The data representation model used in AEA is bag-of-annotations (Fig. 2). By analogy with a bag-of-words representation used in text mining, bag-of-annotations associates a set of annotation terms with each gene (Beißbarth and Speed, 2004; Berriz *et al.*, 2003; Zhang *et al.*, 2005; Zhou and Su, 2007). While bag-of-annotations is a very popular and efficient model allowing natural application of statistical inference methods, it has a number of disadvantages. The main weakness of the model is the limitation in the types of annotation terms and relations that may be used as well as the types and the complexity of enriched phenomena that can be discovered and described.

Several research projects have proposed improvements to AEA algorithms and statistical models to address the issues partially rooted in the bag-of-annotation model. For example, graph decorrelation methods (Alexa *et al.*, 2006; Bauer *et al.*, 2008; Grossmann *et al.*, 2007) modify statistical tests to consider the GO graph; ProfCom method (Antonov *et al.*, 2008) finds enriched combinations of annotation terms using *and*, *or*, *not* operators; GSEA method (Subramanian *et al.*, 2005) compares the study set to curated sets; DAVID tool (Huang *et al.*, 2007) partitions a set of genes based on the annotations; CLEAN tool uses cluster analysis of annotations (Freudenberg *et al.*, 2009). However, these are solutions targeting specific databases or output structure.

To overcome the analytical challenges posed by the bag-of-annotations model, we propose a new paradigm: Annotation Concept Synthesis and Enrichment Analysis (ACSEA). ACSEA utilizes a logic-based data representation model and a fusion of inductive logic reasoning and statistical inference in the general framework of AEA. The results of ACSEA's evaluation suggest that it is a very potent technique capable of increasing the efficiency (i.e. the ease of data analysis by a human expert) and effectiveness (i.e. the quality and quantity of the obtained knowledge) of the processing of high-throughput experiments.

2 METHODS

The cornerstone of ACSEA is a logic-based representation and mining model. In this model, all readily available information about genes is represented by logic statements. Inductive logic reasoning together with statistical inference is then applied to synthesize logic formulas (called annotation concepts) discriminating genes belonging to the study set from genes belonging to the universe set. Then, following AEA approach, constructed annotation concepts are sorted according to their *P*-value and the best of them are presented to the biology expert.

2.1 Logic-based knowledge representation

Due to the evolutionary, distributed and complex nature of biological research, modern biological knowledge is spread over many annotation databases. The captured knowledge itself is of very diverse and often complex structure: Gene Ontology (multiple ontologies/DAGs); KEGG (pathways); InterPro motifs (DAG of sequence patterns); Swiss-Prot keywords (bag of words); PubMed (literature); BIND (interaction map); Protein Databank (sequences, 3D structures, global properties); and UniProt, NCBI (cross-references).

The relational and structured nature of the collected information makes it hard to represent it in the bag-of-annotations model. At the same time, the logic representation, specifically First-Order Logic (FOL), has the following clearly identifiable advantages:

- diverse domain-specific knowledge can be easily integrated with the original data without loss of information;
- any other available knowledge such as conditions of the experiment, constraints, source and reliability of information can be represented and included into the analysis;
- study and universe sets containing compound objects (such as gene–gene interactions or protein complexes) can be naturally portrayed by representing relationships as logic predicates;
- a variety of annotation concepts can be easily described due to the high expressive power of FOL;
- significant amounts of domain-specific knowledge are already captured and formalized as OWL (Web Ontology Language) and RDF (Resource Description Framework) knowledge bases, which are essentially formalisms based on Description Logic, a subset of FOL; and
- obtained annotation concepts can be straightforwardly interpreted by a human expert.

Each fact from the background knowledge, e.g. a gene function, a protein–protein interaction, etc., is transformed into a FOL statement in a form $\texttt{relation\_name(entity}_1\texttt{, entity}_2\texttt{, ..., entity}_n\texttt{)}$. Table 1 illustrates the typical types of knowledge included into analysis. Table 2 shows an example of how the GO structures and the GO gene annotations can be represented by FOL formulas.

2.2 Annotation concept inference

The heart of ACSEA is the Annotation Concept Inference algorithm. The algorithm fuses Inductive Logic Programming (ILP) (Muggleton, 1995; Muggleton and De Raedt, 1994; Page and Srinivasan, 2003) and Statistical Inference (Rivals *et al.*, 2007) approaches.

2.2.1 ILP ILP is an approach to Machine Learning that takes as input a set of positive examples E^+, an optional set of negative examples E^-, background knowledge B, and produces a hypothesis h, such that $B\wedge h \models E^+$, $B\wedge h\wedge E^- \not\models \bot$. All the data in the system (E^+, E^-, B, h) are definite clauses

Table 1. Typical types of background knowledge

Type	Description	Source
Annotations	Annotations are associative relations between objects of interest in a study set and objects in annotation databases. This is the part of knowledge typically covered by the bag-of-annotations representation. For logic-based representation, annotation relations may also include attributes characterizing confidence in the annotation (for not-curated data sources), source of the annotation, etc.	Content of biological databases
Structured background knowledge	Structured background knowledge reflects relations between annotations themselves. Typically, it contains the definition of ontology or a map of annotation terms.	Meta information about a biological database
Expert knowledge	Expert knowledge contains higher level relations about annotation terms and their organization that are not directly expressed by the structured knowledge. For example, for ontology analysis it is customary to add notions such as parent, child, sibling; for a graph, it is neighbor, clique and node distance.	Experts in bioinformatics and biology, published research based on data from biological databases.
Other knowledge	Other knowledge may include information describing phenotypes tested, environmental impact, experimental setup, etc.	Experiment description

Table 2. Logic-based representation of GO annotations

Type	Formula	Comments
Annotations	go_annotation (aah1,go_0005634,c).	The formula states that gene AAH1 is annotated with GO category GO:0005634 from the component ontology.
Structured background knowledge	go_is_a (go_0044424,go_0044464). go_part_of(go_0044424,go_0005622).	The formulas define relations between GO categories. The whole GO direct acyclic graph can be represented in such way.
Expert knowledge	go_anc(A,P) :- go_is_a (A,P). go_anc (A,P) :- go_is_a (A,X), go_anc(X,P). go_sibling(A,B) :- go_is_a(A,P), go_is_a(B,P).	The formulas define useful relations on a graph such as ancestor and sibling.

of the following form: $h \leftarrow b_1, b_2, \ldots, b_n$ where $h, b_1, b_2, \ldots, b_n$ are atoms. As E^+ and E^- represent examples, they usually are ground clauses.

An ILP algorithm constructs a theory in a greedy fashion, adding hypotheses one by one. Typically an ILP algorithm consists of the following sequence of steps (Srinivasan, 2007):

1. Select an example from the E^+ set.
2. Using the background knowledge B, build the most specific clause describing the selected example.
3. Try to generalize the most specific clause (do a search in a clause lattice formed by the most specific clause and an empty clause). If a generalized clause that meets fitness criteria is found (with respect to E^+ and E^- coverage), add it to the theory.
4. Remove from E^+ all examples covered by the generalized clause and repeat from step 1.

2.2.2 Statistical inference Statistical Inference relies on the statistical hypothesis testing methodology (specifically on null-hypotheses tests) to detect a significant enrichment of annotation terms. Generally, the null-hypothesis test consists of the following steps:

1. Define a null hypothesis H_0, which we will try to disprove during the test. The null hypothesis is selected to contrast the tested (alternative) hypothesis H_1. For Enrichment Analysis, the null hypothesis usually states that the property of a gene to have specific annotation and its property to belong to the study set are independent. The tested hypothesis states that these properties are dependent and thus the annotation has different distributions in the study set and in the universe set.
2. Select a statistic that will be used to test hypothesis. For Enrichment Analysis, it is the number of times an annotation appears in the study set.
3. Assuming that the null hypothesis is correct, compute the probability (P-value) of observing a value for the test statistic that is as extreme or more extreme as the value that was actually observed. For Enrichment Analysis, this step relies on the universe set and the statistical distribution model (such as hypergeometric distribution) to compute the probability.
4. Based on the computed P-value, the null hypothesis can be rejected if the P-value falls below a significance threshold (critical value).

2.2.3 Annotation concept inference By fusing the inductive logic reasoning and the statistical inference approaches we obtain an inference algorithm capable of mining complex knowledge structures while tolerant to noise and data incompleteness.

While several probabilistic/logic inference models exist (such as Probabilistic Inductive Logic Programming), they incorporate statistical information directly into the produced hypotheses. As a result, they are very potent models for classification problems; however, they significantly diminish the key advantage of logic-based approaches, namely the human understandability of generated hypotheses. Hypotheses produced by PILP are complex Bayesian networks with attached sets of probability tables (see Figure 10.4 in Kersting and De Raedt, 2007). Therefore, they are not well suited for the explanatory type of analysis ACSEA is performing.

The ACSEA approach consists of the following key elements: annotation concept synthesis, a hypothesis fitness measure, a theory building strategy, integration of specialized algorithms, and methods for controlling the quality of the theory.

Annotation concept synthesis: ACSEA processes experimental data by synthesizing relevant annotation concepts. Annotation concepts are logic formulas that capture discriminating information about the study and universe sets. The concepts are synthesized following the Inductive Logic Programming framework. E^+ is populated from the study set. E^- is populated by the universe set less the study set. Hypotheses constructed during the inference process, by the design of the system (see below), correspond to the annotation concepts that capture discriminating knowledge.

Hypothesis fitness measure: the hypothesis fitness measure guides the hypothesis generalization search in the clause lattice and is used to compare and select the best hypothesis. ILP classification systems typically employ accuracy, entropy, coverage, or similar measures. ACSEA applies statistical hypothesis testing based on the hypergeometric model as its hypothesis fitness measure.

$$P(t) = P(X \geq n_t) = \sum_{i=n_t}^{\min(n,m_t)} \frac{\binom{m_t}{i}\binom{m-m_t}{n-i}}{\binom{m}{n}} \qquad (1)$$

where $P(t)$ is the P-value of enrichment of annotation t in the study set S, according to the one-sided hypergeometric test; $n, m,$ are the sizes of the universe and the study sets, n_t, m_t are the numbers of genes annotated by t in the universe and the study sets, respectively.

The hypothesis fitness measure is also consulted to prune parts of the search space. The pruning decision is based on the possibility of finding a hypothesis that either (i) is better than the best hypothesis found so far, or (ii) has the fitness above a predefined threshold. The ACSEA system includes an estimation algorithm to compute the hypotheses fitness bounds for parts of the search space based on the statistical hypothesis testing measure.

Theory building strategy: ILP classification systems typically build theory according to one of the following greedy strategies: induction of minimal covering theory, induction of maximal theory, or feature construction. The goal of the first two strategies is to find a fairly limited number of hypotheses covering all examples. Such strategies are not particularly suited for ACSEA as in most cases no complete trustworthy coverage exists due to the noise in experimental data as well as the noise and omissions in background knowledge. Furthermore, one example may potentially lead to several significantly different annotation concepts. The goal of the last strategy is to find all (potentially a very high number) of hypotheses that meet the fitness criteria. Such type of strategies would generate an overwhelming amount of hypotheses.

The most natural goal for an ACSEA-specific theory building strategy is to find a limited number of the highest quality hypotheses. To meet this goal, ACSEA defines a sliding-window theory building strategy. During the search, a fixed size set of hypotheses meeting the fitness criteria is maintained. When a better hypothesis is found, it's added to the set, while the worst hypothesis in the set is removed. The fitness criteria are revised to a higher standard as a result.

Such approach has a 2-fold advantage:

(1) at the end of the search, the theory contains a predefined number of the highest quality hypotheses;

(2) the efficiency of the fitness-based search space pruning is constantly increases during the search.

In the preliminary experiments, we compared the performance of ACSEA with and without the sliding window strategy on several datasets. The sliding window ACSEA version outperformed the feature construction ACSEA version on all datasets according to all used measured (see Section 3.1) with confidences >90%.

Integration of specialized algorithms: a significant advantage of the logic-based systems is their ability to integrate external specialized data mining algorithms. Originally, such integration was proposed to make ILP systems capable of performing numerical data analysis (Srinivasan and Camacho, 1999). In the bioinformatics context, the same approach can be used to process gene's quantitative properties, sequences, keywords, etc.

The external algorithms are inserted into an ILP system as a special kind of predicates (lazily evaluable predicates implementing learning and classification forms of execution). During the hypothesis search, they are inserted into the clause as one of the atoms and the underlying algorithm is invoked.

To validate the usefulness of such integration in ACSEA, we implemented a statistical pattern recognition algorithm that compares the distributions of numerical attributes based on the one-dimensional Gaussian model. ACSEA successfully applied the algorithm to model the gene distributions along chromosomes (Fig. 3).

Controlling the quality of the theory: as the theory built by ACSEA is going to be presented and evaluated by human experts, a number of measures have been incorporated into the algorithm to control the objective and subjective quality of the theory, i.e. its size, redundancy, readability and understandability.

The theory building strategy selects a predefined number of the highest quality hypotheses evaluated with a hypothesis fitness measure. This measure includes quantitative thresholds such as the maximum P-value and minimal positive example coverage as well as qualitative requirements to hypotheses.

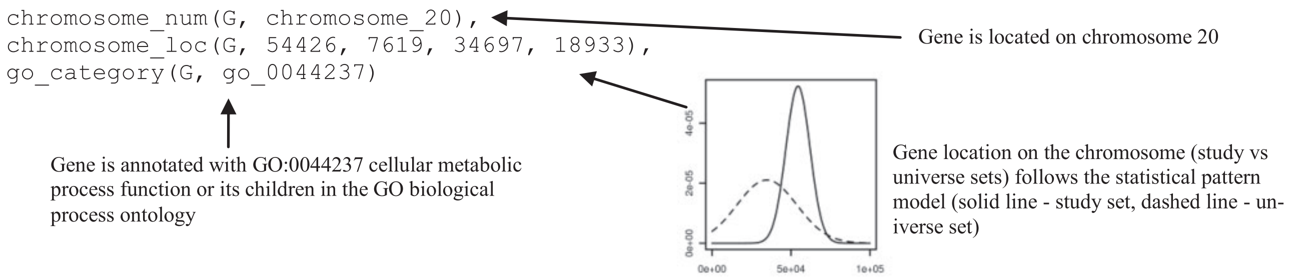
The qualitative requirements are stated as syntactic integrity constraints that are used to discard individual hypotheses or prune parts of the lattice. Meta-information contained in annotation datasets is one source of the integrity constraints (so they are part of the background knowledge). The integrity constraints can also be provided by a biology expert, when they are dictated by the expert's area of research or conditions of the experiments. For example, one may restrict hypotheses to have no more than one reference to each ontology from GO.

Another technique to improve the quality of a theory is filtering out highly overlapping (synonymic) hypotheses. A high number of synonymic hypotheses is a natural consequence of the abundance of alternative terms in the background knowledge and multiple ways of expressing essentially the same hypothesis by the FOL formulas. Currently, we utilize an algorithm that assesses the hypotheses based on their coverage of the study and universe sets and discards more complex hypotheses having the identical coverage.

2.3 Search space tractability

A weakness shared by algorithms based on Inductive Logic Programming is a large search space and fairly large computational requirements for the evaluation of hypotheses. The size of the search space can be roughly estimated as $\binom{a}{r}$, where r is the maximum number of atoms in considered hypotheses, and a is the number of statements in the background knowledge that can be used as atoms for hypotheses. For example, for one of the problems considered in this article, the full search space roughly contains $\binom{17000}{3} \cong 10^{12}$ hypotheses. We address the tractability issue with the following countermeasures integrated into our approach:

- Utilize strict hypotheses fitness criteria allowing to significantly prune the search space.
- Assert constraints on the size of hypotheses (the number of atoms in a logic formula). A human expert should be able to quickly assess the hypotheses, so limiting the size of hypotheses helps to reduce the search space and improve the quality of the results.
- Integration of specialized algorithms. Specialized algorithms to analyze specific data types can outperform a logic program as many algorithms are more efficient when implemented in imperative or functional paradigms.
- Pruning of the background knowledge. The background knowledge may be pruned if it contains no information that can be referred to from the study and universe sets. The pruning is performed when data are converted from biomedical databases to a logic representation.



```
chromosome_num(G, chromosome_20),
chromosome_loc(G, 54426, 7619, 34697, 18933),
go_category(G, go_0044237)
```

Gene is located on chromosome 20

Gene is annotated with GO:0044237 cellular metabolic process function or its children in the GO biological process ontology

Gene location on the chromosome (study vs universe sets) follows the statistical pattern model (solid line - study set, dashed line - universe set)

Fig. 3. An example of a synthesized annotation concept. *chromosome_num* predicate describes the relation between a gene and a chromosome, *chromosome_loc* tests the location of the gene on a chromosome against a learned model (the location is specified in base pairs, the predicate parameters are populated with the mean and variance of two normal distributions modeling the study and universe sets), *go_category* specifies the relation between a gene and a GO term.

Table 3. Microarray datasets

Dataset name	Dataset description
Bioconductor ALL	Data of T- and B-cell acute lymphocytic LEUKEMIA from the Ritz Laboratory at the DFCI
GSEA gender	Transcriptional profiles from male and female lymphoblastoid cell lines
GSEA p53	Transcriptional profiles from p53+ and p53 mutant cancer cell lines
GSEA diabetes	Transcriptional profiles of smooth muscle biopsies of diabetic and normal individuals
GSEA leukemia	Transcriptional profiles from leukemias—0ALL and AML
GSEA lung cancer	Transcriptional profiles from lung cancer outcome datasets

3 RESULTS

We evaluated the performance of ACSEA[1] applied to two widely used experimental techniques: expression microarrays and genetic interaction screens.

This section contains summary of the performed experiments. More detailed experimental logs and raw results are available at ACSEA home page (http://www.epiphan.com/~zhilin/ACSEA).

3.1 Expression microarrays

To evaluate the performance of ACSEA, we selected six well-known microarray datasets listed in Table 3.

For each dataset, we applied nonspecific filtering, removing genes having inter-quartile range less than 0.5. Such filtering leaves only genes with sufficient variability to be informative. Next, we applied the standard t-test with the P-value threshold of 0.05 to identify differentially expressed genes. Differentially expressed genes formed the study set, while all genes left after the nonspecific filtering formed the universe set. Further, for each individual experiment, depending on the used annotation database, genes without any annotation attached were removed from both sets. Each dataset was analyzed with the annotation sources listed in Table 4.

Table 4. Annotation databases for microarray datasets

Name	Description
GO	Gene ontology, released October 2009 Gene ontology annotation for human, released October 2009
GCM (gene to chromosome mapping)	Gene to chromosome mapping (chromosome, chromosome band, and start/end base pairs) from Ensembl 56 database
GO + GCM	Combination of the two annotation sources above

We defined a family of performance measures to evaluate the proposed approach. The measures are based on assessing the P-values for the set of generated hypotheses. The main idea is that after ‘synonymic’ hypotheses are removed, we would like to minimize the P-value of several top hypotheses.

$$PvAvr_n(T) = \frac{1}{n}\sum_{i=1}^{n} Pvalue(\mathrm{t_i}) \qquad (2)$$

where T is a theory consisting of a list of hypotheses $\{t_i\}$ sorted in ascending order by their P-values, and n is the number of the top hypotheses included into evaluation. The similar measures can be defined for P-values adjusted for multiple testing.

$$QvAvr_n(T) = \frac{1}{n}\sum_{i=1}^{n} Qvalue(\mathrm{t_i}) \qquad (3)$$

In our work we used Bonferroni correction, which is the strictest way of addressing the problem of multiple testing.

$$Qvalue(\mathrm{t_i}) = ||T||P\mathrm{value}(\mathrm{t_i}) \qquad (4)$$

where $||T||$ represents the total number of unique tested hypothesis.

Parameter n in $PvAvr_n(T)$ and $QvAvr_n(T)$ allows to evaluate the quality of theories of different sizes produced by an algorithm. As a biological experiment may capture several phenomena, it is reasonable to expect that the enrichments theory will include more

[1] The ACSEA system described in this article was implemented based on R, a statistical computation system (http://www.r-project.org/); Bioconductor, a system for the analysis and comprehension of genomic data (http://www.bioconductor.org/); YAP, a high-performance Prolog compiler (http://www.dcc.fc.up.pt/~vsc/Yap/); and Aleph, an ILP system (http://web.comlab.ox.ac.uk/activities/machinelearning/Aleph/). The system is available at http://www.epiphan.com/~zhilin/ACSEA

Table 5. Quantitative performace evaluation of AEA and ACSEA on gene expression microarray datasets

Dataset	Annotations	QvAvr_1		QvAvr_5		QvAvr_{10}		QvAvr_{25}	
		AEA	ACSEA	AEA	ACSEA	AEA	ACSEA	AEA	ACSEA
ALL	GO	6.33e-02	**4.48e-04**	1.45e-01	**6.85e-04**	3.41e-01	**1.06e-03**	7.36e-01	**2.89e-03**
	GCM	4.55e-01	**3.37e-01**	8.91e-01	**6.61e-01**	9.45e-01	**8.30e-01**	9.78e-01	**9.32e-01**
	GO + GCM	1.30e-01	**1.19e-03**	2.33e-01	**1.63e-03**	5.88e-01	**3.81e-03**	8.35e-01	**8.07e-03**
GSEA Gender	GO	8.25e-04	**1.86e-05**	2.29e-02	**5.13e-05**	2.76e-01	**6.79e-05**	7.11e-01	**1.14e-04**
	GCM	1.25e-04	**5.35e-05**	2.22e-01	**9.86e-03**	6.11e-01	**1.18e-01**	8.44e-01	**6.31e-01**
	GO + GCM	8.12e-04	**8.13e-05**	6.38e-03	**9.92e-05**	7.08e-02	**1.45e-04**	6.05e-01	**8.17e-04**
GSEA p53	GO	1.00e + 00	**1.39e-01**	1.00e + 00	**2.11e-01**	1.00e + 00	**2.79e-01**	1.00e + 00	**4.64e-01**
	GCM	6.23e-04	**3.93e-05**	6.84e-01	**7.27e-02**	8.42e-01	**2.10e-01**	9.37e-01	**5.82e-01**
	GO + GCM	1.76e-02	**6.18e-03**	8.04e-01	**3.18e-02**	9.02e-01	**7.16e-02**	9.61e-01	**2.07e-01**
GSEA Diabetes	GO	1.00e + 00	**2.89e-01**	1.00e + 00	**6.96e-01**	1.00e + 00	**8.48e-01**	1.00e + 00	**9.39e-01**
	GCM	1.00e + 00	**1.40e-01**	1.00e + 00	**4.51e-01**	1.00e + 00	**7.26e-01**	1.00e + 00	**8.90e-01**
	GO + GCM	1.00e + 00	**3.61e-01**	1.00e + 00	**6.80e-01**	1.00e + 00	**8.40e-01**	1.00e + 00	**9.36e-01**
GSEA Leukemia	GO	**4.80e-02**	2.48e-01	5.51e-01	**2.83e-01**	7.76e-01	**3.40e-01**	9.10e-01	**4.96e-01**
	GCM	6.49e-01	**6.28e-01**	9.30e-01	**8.69e-01**	9.65e-01	**9.35e-01**	9.86e-01	**9.74e-01**
	GO + GCM	**8.39e-02**	3.25e-01	6.81e-01	**3.69e-01**	8.41e-01	**4.35e-01**	9.36e-01	**6.18e-01**
GSEA Lung Cancer	GO	2.67e-01	**1.37e-01**	8.53e-01	**3.48e-01**	9.26e-01	**4.22e-01**	9.71e-01	**5.85e-01**
	GCM	1.79e-05	**6.17e-06**	5.41e-01	**9.02e-02**	7.70e-01	**2.69e-01**	9.08e-01	**6.92e-01**
	GO + GCM	5.29e-04	**2.07e-04**	6.55e-01	**2.23e-04**	8.28e-01	**2.50e-04**	9.31e-01	**4.45e-04**

Lesser is better (better values are highlighted). The differences in performance are statistically significant with 95% (n = 1) and 99% (n = 5, 10, 25) confidence levels.

than one interpretation. At the same time, the size of a theory must be limited by a number of enrichments a biology expert can comfortably assess in an allocated time frame. In our work we used several values of n ranging from 1 to 25.

Theoretically, the P-value based performance results of ACSEA cannot be worse than results of AEA for the same task. It is because the hypothesis space of ACSEA contains the hypothesis space of AEA. However, it is true only if both algorithms exhaustively walk through their search spaces. For ACSEA, in the presence of significant amounts of background knowledge, the exhaustive search is intractable in practice. Thus, the P-value based performance measure helps to evaluate the algorithms as a whole, including the crucial optimizations that made ACSEA feasible for practical use. Moreover, in our analysis we mostly use Q-value based measures. Q-values are derived from P-values by applying a Multiple Testing Correction (MTC) procedure. MTC penalizes an algorithm for each additional statistical test it performs (such as evaluation of a hypothesis). Therefore, the Q-value is a performance measure that compares algorithms on one scale regardless of the sizes of their hypothesis spaces.

We compared the ACSEA approach to the AEA approach represented by Bioconductor's Category/GOstats algorithm. The Category and GOstats packages were extended to analyze arbitrary annotations so the algorithms can be compared on a variety of different annotation sources. The same final statistical analysis was followed to calculate the P-values for enriched annotations discovered by AEA and ACSEA. Consequently, the Bonferroni correction was applied to address the problem of multiple comparisons. The family of measures was transformed accordingly.

The algorithms produced theories consisting of lists of annotation terms for AEA or logic formulas for ACSEA. The quantitative performance evaluation results[2] are presented in Table 5. In a majority of cases the ACSEA approach suggested annotations at least one (often two and three) order of magnitude better than the bag-of-annotations based AEA algorithm. The quality of constructed annotations and the level of the integrative information analysis performed by ACSEA can be illustrated by the enriched annotation discovered during the Diabetes/GO+GCM experiment (Fig. 3).

3.2 Protein and genetic interaction screens

The ACSEA approach was evaluated on the DRYGIN (Data Repository of Yeast Genetic Interactions) dataset (Costanzo *et al.*, 2010; Koh *et al.*, 2010). DRYGIN contains genetic interaction results for 1712 × 3885 tested pairs of genes. The raw results are grouped into the stringent-cutoff, intermediate-cutoff and lenient-cutoff datasets based on the strength of the evidence supporting detected interactions. The stringent-cutoff dataset contains only interactions having strong experimental support, while lenient-cutoff dataset includes interaction supported by weaker evidences. The stringent-cutoff dataset was selected for ACSEA evaluation.

The gene interaction information from DRYGIN was converted to a symmetric Boolean matrix. Two-dimensional clustering algorithms were applied to the matrix. Two clustering algorithms were selected: PAM (Partitioning Around Medoids) and K-means (Hartigan and Wong variant).

[2] Generally, ACSEA is a slower algorithm than AEA in approximately 10–100 times depending on the task. We did not include formal comparison as our implementation of ACSEA is not optimized on programming language, code and compiler levels. The speed may be improved by switching from the interpretation of Prolog to the compilation of Prolog or the use of C/C++ code.

The list of detected clusters was filtered by removing clusters that include less than 25 genes or more than 33% of all genes, or have less than 25 intra-cluster interactions. Then, for each cluster an independent ACSEA and AEA experiment was performed, where the set of intra-cluster interactions formed a study set while all interactions formed the universe set.

Each experiment was carried out with the annotation sources listed in Table 6.

For ACSEA, annotations for each interacting pair of genes were converted to logic statements. For the AEA algorithm, for each interaction the set of annotation terms $\{a_i\}$ was obtained by taking the union of the two sets of annotation terms describing interacting genes $\{a_i^1\}$ and $\{a_i^2\}$.

$QvAvr_n$ measures were computed for each experiment for the ACSEA and AEA algorithms. The quantitative performance evaluation results are presented in Table 7. The quality of constructed annotations and the types of structural analysis performed by ACSEA can be illustrated by a synthesized concept in Figure 4.

3.3 Results summary

Tables 5 and 7 show that ACSEA outperforms the AEA algorithm in almost all cases, even when the most conservative Bonferroni correction is applied. ACSEA tends to exhibit the best results on larger, well-structured annotation sets (such as GO+GCM in our tests), which is expected (as Inductive Logic Programming relies on the rich background knowledge) and welcomed (as the most help is needed from annotation enrichment tools in such cases).

ACSEA demonstrated (Fig. 3) that more complex and advanced (comparing to AEA's treatment of numerical annotations as nominal values) analysis of numerical annotations can be performed as an integral part of the enrichment analysis. In a similar manner, the enrichment analysis can be extended to directly operate on any data type essential for an experiment such as strings, sequences, vectors, etc.

ACSEA was capable of uncovering enriched phenomena tied to the structure of the analyzed objects (Fig. 4). That, by itself, opens a new dimension in the enrichment analysis. Such analysis is likely to be even more important for sets of yet more complex objects than gene interactions (protein complexes, for example).

The obtained results suggest that ACSEA boosts the efficiency and effectiveness of the processing of high-throughput experiments such as expression microarrays and genetic interaction screens by finding better, more integrative interpretations of biological experiments.

Table 6. Annotation databases for genetic interaction screens

Name	Description
GO	GO annotations. Bioconductor GO.db package, version 2.2.11. Bioconductor org.Sc.sgd.db package, version 2.2.12
GCM (Gene to chromosome mapping)	Gene to chromosome mapping. Bioconductor org.Sc.sgd.db package, version 2.2.12
GO + GCM	Combination of the two annotation sources above

4 DISCUSSION

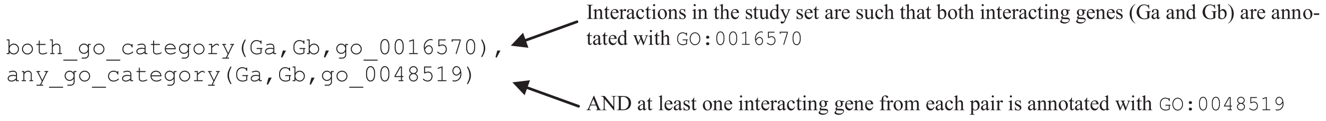
AEA is becoming the dominant technique for the secondary processing of data generated by high-throughput experimental techniques. Significant progress in AEA algorithms has been obtained by improving the statistical models and by incorporating a variety of annotation databases into the analysis. In this article, we present a novel paradigm, ACSEA, which relies on a logic-based representation of annotations and employs a fusion of inductive logic inference and statistical inference.

The methodological advantage of ACSEA is 5-fold. First, it is easier to represent complex, structural annotation information.

Table 7. Quantitative performance evaluation of AEA and ACSEA on genetic interaction screens

Dataset	Annotations	$QvAvr_1$		$QvAvr_5$		$QvAvr_{10}$		$QvAvr_{15}$	
		AEA	ACSEA	AEA	ACSEA	AEA	ACSEA	AEA	ACSEA
PAM	GO	8.27e-09	**8.33e-10**	5.70e-06	**1.42e-09**	1.20e-04	**1.85e-09**	2.75e-04	**3.24e-09**
	GCM	**1.09e-06**	6.08e-06	2.88e-01	**2.30e-03**	6.44e-01	**2.64e-02**	7.63e-01	**9.43e-02**
	GO + GCM	8.31e-09	**3.32e-10**	3.20e-06	**1.24e-09**	9.20e-05	**1.37e-08**	2.28e-04	**3.10e-08**
K-means	GO	7.36e-06	1.42e-06	3.50e-04	**2.10e-06**	1.66e-03	**3.81e-06**	2.68e-03	**5.38e-06**
	GCM	1.12e-02	1.66e-02	3.81e-01	**1.14e-01**	6.79e-01	**1.88e-01**	7.86e-01	**2.88e-01**
	GO + GCM	2.59e-07	**3.37e-09**	3.41e-07	**6.51e-09**	2.12e-06	**1.52e-08**	2.14e-05	**2.11e-08**

Lesser is better (better values are highlighted). The differences in performance are statistically significant with 99% ($n=5, 10, 15$) confidence level.



```
both_go_category(Ga,Gb,go_0016570),
any_go_category(Ga,Gb,go_0048519)
```

Interactions in the study set are such that both interacting genes (Ga and Gb) are annotated with GO:0016570

AND at least one interacting gene from each pair is annotated with GO:0048519

Fig. 4. An example of a synthesized annotation concept. Preservation of the internal structure of object 'genetic interaction' allows reasoning on and inferring statements dealing with the substructure of the object under analysis. Particularly, in this experiment we were able to utilize quantifiers such as *both…*, *any…*, *one…* to better understand relationship between GO categories and the set of interacting genes.

Information already captured and formalized in OWL and RDF knowledge bases can be directly utilized. Secondly, it is possible to synthesize and analyze complex annotation concepts. Thirdly, it is possible to perform the enrichment analysis for sets of aggregate objects (such as sets of genetic interactions, physical protein–protein interactions or sets of protein complexes). Fourthly, annotation concepts are straightforward to interpret by a human expert. Fifthly, the logic data model and logic induction are a common platform that can integrate specialized analytical tools.

We evaluated ACSEA on several microarray and genetic interaction datasets. Our results demonstrate that the proposed approach synthesizes higher quality integrated interpretation of biological phenomena captured by biological experiments.

The work presented here can also be viewed as an innovative application of the ILP theory. While normally ILP techniques are used for classification tasks involving relational data, this research shows how an approach, incorporating inductive logic ideas, can serve as a knowledge integration mechanism, enriching the data with relational background knowledge and resulting in comprehensible interpretations of the experimental data.

For future work, we plan to pursue the following research directions. First, we will advance the theory consolidation algorithm that can remove 'synonymic' annotations based on coverage and elements of theorem proving. Secondly, inductive annotation construction can be extended by abduction (Kakas *et al.*, 1993) to immediately suggest annotations missing in the annotation databases that can be directly inferred based on new experimental data. Thirdly, we are going to investigate the possibility of using Description Logic as a more efficient alternative to FOL.

Conflict of Interest: none declared.

REFERENCES

Alexa,A. *et al.* (2006) Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*, **22**, 1600–1607.

Alibés,A. *et al.* (2008) PaLS: filtering common literature, biological terms and pathway information. *Nucleic Acids Res.*, **36**, W364–W367.

Al-Shahrour,F. *et al.* (2004) FatiGO: a web tool for finding significant associations of gene ontology terms with groups of genes. *Bioinformatics,* **20**, 578–580.

Antonov,A. *at al.* (2008) ProfCom: a web tool for profiling the complex functionality of gene groups identified from high-throughput data. *Nucleic Acids Res*, **36**, W347–W351.

Bauer,S. *et al.* (2008) Ontologizer 2.0 - a multifunctional tool for GO term enrichment analysis and data exploration. *Bioinformatics,* **24**, 1650–1651.

Beißbarth,T. and Speed T.P. (2004) GOstat: find statistically overrepresented gene ontologies within a group of genes. *Bioinformatics*, **20**, 1464–1465.

Berriz,G.F. *et al*. (2003) Characterizing gene sets with FuncAssociate. *Bioinformatics*, **19**, 2502–2504.

Costanzo,M. *et al*. (2010) The genetic landscape of a cell. *Science*, **327**, 425–431.

Freudenberg,J.M. *et al.* (2009) CLEAN: CLustering Enrichment ANalysis. *BMC Bioinformatics*, **10,** 234.

Grossmann,S. *et al.* (2007) Improved detection of overrepresentation of gene-ontology annotations with parent-child analysis. *Bioinformatics,* **23**, 3024–3031.

Huang,D.W. *et al.* (2009) Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.*, **37**, 1–13.

Huang,W. *et al.* (2007) The DAVID gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists. *Genome Biol.,* **8**, R183.

Kakas,A.C. *et al.* (1993) Abductive logic programming. *J. Logic Comput.*, **2**, 719–770.

Khatri,P. and Draghici,S. (2005) Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, **21**, 3587–3595.

Koh,J. *et al.* (2010) DRYGIN: a database of quantitative genetic interaction networks in yeast. *Nucleic Acids Res.*, **38**, D502–D507.

Kersting,K. and De Raedt,L. (2007) Bayesian logic programming: theory and tool. In Getoor L. and Taskar B. (eds.) *Introduction to Statistical Relational Learning*. The MIT Press, pp. 291–321.

Minguez,P. *et al.* (2007) Functional profiling of microarray experiments using text-mining derived bioentities. *Bioinformatics,* **23**, 3098–3099.

Muggleton,S.H. (1995) Inverse entailment and progol. *New Gen. Comput.*, **13**, 245–286.

Muggleton,S.H. and De Raedt,L. (1994) Inductive logic programming: theory and methods. *J. Logic Program.,* **19–20**, 629–679.

Page,D. and Srinivasan,A. (2003) ILP: a short look back and a longer look forward. *J. Mach. Learn. Res.*, **4**, 415–430.

Ramakrishnan,N. *et al.* (2005) Reconstructing formal temporal models of cellular events using the GO process ontology. In *Proceedings of Bio-Ontologies SIG Meeting*. Bio-Ontologies SIG Meeting, ISMB 2005, Detroit, USA.

Rivals,I. *et al.* (2007) Enrichment or depletion of a GO category within a class of genes: which test? *Bioinformatics,* **23**, 401–407.

Sherman,B. *et al.* (2007) DAVID knowledgebase: a gene-centered database integrating heterogeneous gene annotation resources to facilitate high-throughput gene functional analysis. *BMC Bioinformatics*, **8**, 426.

Srinivasan,A. (2007) The Aleph manual: version 4 and above.

Srinivasan,A. and Camacho,R. (1999) Numerical reasoning with an ILP system capable of lazy evaluation and customised search. *J. Logic Program.*, **40**, 185–213.

Subramanian,A. *et al.* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci.*, **102**, 15545–15550.

Zhang,B. *et al.* (2005) WebGestalt: an integrated system for exploring gene sets in various biological contexts. *Nucleic Acids Res.*, **33**, W741–W748.

Zhou,X. and Su (2007) EasyGO: Gene ontology-based annotation and functional enrichment analysis tool for agronomical species. *BMC Genomics*, **8**, 246.