BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 19 2005, pages 3787–3793
doi:10.1093/bioinformatics/bti430

Databases and ontologies

Automated genome annotation and pathway identification using the KEGG Orthology (KO) as a controlled vocabulary

Xizeng Mao[1,†], Tao Cai[1,†], John G. Olyarchuk[1], and Liping Wei[1,2,*]

[1]Center for Bioinformatics, National Laboratory of Protein Engineering and Plant Genetic Engineering, College of Life Sciences, Peking University, Beijing 100871, P.R. China and [2]Biomedical Informatics, Department of Medicine, Stanford University School of Medicine, Stanford, CA 94305, USA

Received on January 18, 2005; revised on March 26, 2005; accepted on April 4, 2005
Advance Access publication April 7, 2005

ABSTRACT

Motivation: High-throughput technologies such as DNA sequencing and microarrays have created the need for automated annotation of large sets of genes, including whole genomes, and automated identification of pathways. Ontologies, such as the popular Gene Ontology (GO), provide a common controlled vocabulary for these types of automated analysis. Yet, while GO offers tremendous value, it also has certain limitations such as the lack of direct association with pathways.

Results: We demonstrated the use of the KEGG Orthology (KO), part of the KEGG suite of resources, as an alternative controlled vocabulary for automated annotation and pathway identification. We developed a KO-Based Annotation System (KOBAS) that can automatically annotate a set of sequences with KO terms and identify both the most frequent and the statistically significantly enriched pathways. Results from both whole genome and microarray gene cluster annotations with KOBAS are comparable and complementary to known annotations. KOBAS is a freely available standalone Python program that can contribute significantly to genome annotation and microarray analysis.

Availability: Supplementary data and the KOBAS system are available at http://genome.cbi.pku.edu.cn/download.html

Contact: weilp@mail.cbi.pku.edu.cn

INTRODUCTION

In recent years, high-throughput technologies such as DNA sequencing and microarrays have created the need for the automated annotation and analyses of large sets of genes, including whole genomes. To this end, an ontology, which is defined as a specification of a conceptualization, provides a common controlled vocabulary to facilitate electronic communication and sharing of information across different research groups and enables comparison of annotations across different genomes and different gene sets.

Several ontologies have been developed for genome annotation and expression analysis such as the Gene Ontology (GO) (Ashburner *et al*., 2000), GeneX (http://www.ncgr.org/genex/) and eVOC (Kelso *et al*., 2003). One of the most widely used ontologies is the Gene Ontology (GO). The Gene Ontology organizes functional terms into three top-level categories: molecular function, biological process and cellular component. Each category is structured as a directed acyclic graph (DAG) in which a term may have more than one parent and more than one child. The Gene Ontology has been used in the annotation of many genome databases, including SGD, CGD, FlyBase, MGI, TAIR, ZFIN, DictyBase, WormBase and RGD. Researchers annotating these databases use a combination of automation and manual curation to assign GO terms to genes in these genomes. Computational tools that have been developed include OntoBlast and Goblet, which assign GO terms to a new sequence based on its similarity (often measured by BLAST E-values) to a sequence with a known GO assignment (Hennig *et al*., 2003; Zehetner, 2003), and InterPro, which assigns GO terms to a new sequence based on the known GO assignments of the functional domains identified in the sequence (Mulder *et al*., 2003).

Other tools have been developed to discover significantly enriched GO terms among a given set of sequences such as a set of genes found to form a cluster in a microarray analysis. They include the web-based applications FatiGO (Al-Shahrour *et al*., 2004), GFinder (Masseroli *et al*., 2004), Gostat (Beissbarth and Speed, 2004), NetAffx GO Mining Tool (Cheng *et al*., 2004a), Onto-Express (Draghici *et al*., 2003), the JAVA application GoMiner (Zeeberg *et al*., 2003), the R package OntologyTraverser (Young *et al*., 2005) and the Perl command-line tools GeneMerge (Castillo-Davis and Hartl, 2003) and TermFinder (Boyle *et al*., 2004). These tools calculate the p-value for each GO term seen in the given set of sequences using one or more of the following statistical methods: binomial distribution, hypergeometric distribution, Fisher's exact test and/or chi-square test, using either the entire probe set on the microarray or the complete genome sequence for the background distribution. They also often apply one or more multiple hypotheses correction strategies, such as Bonferroni correction, to control the false discovery rate (FDR). Two other methods have used a knowledge-based approach (Cheng *et al*., 2004b) or DAG (directed acyclic graph) structure (Lee *et al*., 2004) to find meaningful GO terms.

While GO offers tremendous value, it also has certain limitations. Firstly, the GO hierarchy has highly varied depths along different branches—from two levels (e.g. GO:0001662 behavioral fear response) to 15 levels (e.g. GO:0030607 mitotic spindle orientation). Some of the variation is inherent in different functional families, while some may be an artifact of the uneven contribution by different groups participating in GO's development and may affect the reliability of statistical significance tests of GO terms.

[*]To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

Table 1. Examples of genes that had no annotation in KEGG but similar annotations by KOBAS and in SGD (the *Saccharomyces* genome database)

Gene ID	Annotation in KEGG	Annotation by KOBAS	Annotation in SGD
YBL075C	None	K03283, TC.HSP70; heat shock protein 70, Hsp70 family	Heat-inducible cytosolic member of the 70 kDa heat shock protein family
YCR068W	None	K01046, E3.1.1.3; triacylglycerol lipase	Lipase, required for intravacuolar lysis of autophagic bodies
YDL160C	None	K01509, E3.6.1.3; adenosinetriphosphatase	Cytoplasmic DexD/H-box helicase, stimulates mRNA decapping
YER103W	None	K03283, TC.HSP70; heat shock protein 70, Hsp70 family	Member of 70 kDa heat shock protein family

Secondly, because GO was originally developed for the annotation of eukaryotic genomes, the functional categorization in GO, and genome annotation using GO, is not as accurate for some prokaryotes as for eukaryotes. Thirdly, because GO terms do not correspond directly to known pathways; it is difficult to identify pathways directly from GO annotations.

We have investigated alternative controlled vocabularies for automated annotation of sets of genes and propose the KEGG Orthology (KO), part of the KEGG suite of resources (Kanehisa, 1997; Kanehisa and Goto, 2000) as such an alternative. KEGG is best known for its large pathway database and KO was developed to integrate pathway and genomic information in KEGG. Historically, enzyme commission (EC) numbers were used to describe common gene products in metabolic pathways. The ortholog identifiers were later introduced to overcome limitations in the enzyme nomenclature. The KEGG Orthology is a further extension of the ortholog identifiers, and is structured as a DAG hierarchy of four flat levels. The top level consists of the following five categories: metabolism, genetic information processing, environmental information processing, cellular processes and human diseases. The second level divides the five functional categories into finer sub-categories. The third level corresponds directly to the KEGG pathways, and the fourth level consists of the leaf nodes, which are the functional terms.

Our research demonstrates that KO is effective as a controlled vocabulary for automated annotation of sets of sequences, including whole genomes, and since KO links directly to known pathways, KO annotations enable concurrent pathway identification. Surprisingly, there are few published investigations using KO for automated annotations of new sequences, and the only existing tool is GFIT (Bono *et al*., 1998), which assigns EC numbers to query sequences based on orthologous genes in *KEGG GENES*. Unfortunately, the available GFIT package is obsolete and KO is significantly more complicated than EC numbers.

Several existing tools use the KEGG pathway database to identify enriched pathways in microarray data. PathProcessor (Grosu *et al*., 2002) and PathMAPA (Pan *et al*., 2003) map genes to KEGG pathways through their enzyme nomenclature, while ArrayXPath (Chung *et al*., 2004) maps to pathways in KEGG, GenMAPP (Dahlquist *et al*., 2002) and BioCarta (http://www.biocarta.com/), through sequence identifiers. Knowing the EC number or sequence identifier a priori is required to use these tools.

Here we present KOBAS, a KO-Based Annotation System written in Python that uses KO as a controlled vocabulary to automatically annotate, based on BLAST similarity searches, sets of new sequences. KOBAS also identifies the most frequent and the most significantly enriched pathways in a given set of sequences. We demonstrate the effectiveness of KO and KOBAS by evaluating KOBAS on two previously annotated genomes, one newly sequenced genome and gene clusters from a microarray experiment.

METHODS

Selection and parsing of original datasets

We used two datasets available in Release 32.0 of the KEGG suite of resources. First, we extract the KO hierarchy and the known associations between genes and their corresponding KO functional terms from the *KO* data set. The *KO* dataset is a single complex flat file containing entries for all of the KO functional terms (the leaf nodes at the fourth level of the KO hierarchy). An example is shown in Supplementary Table 1. Each entry includes the ID, functional description and hierarchy location of the KO term, links to other databases (e.g. GO, COG, enzyme) and the IDs of genes in all annotated genomes that have this KO function. Second, we extract the amino acid sequences of all the annotated genes from the *KEGG GENES* dataset, which consists of one flat file for each annotated genome of gene entries. Each entry contains the gene's ID, definition, genomic location, KO assignments (if available), and amino acid and nucleotide sequence. An example is shown in Supplementary Table 2. About one-third of all sequences in the genomes in *KEGG GENES* have been assigned KO terms. We observed that while it was possible to extract known associations between genes and KO terms from both the *KO* and the *KEGG GENES* datasets, our detailed comparison of the set of associations extracted from the *KO* dataset versus that from the *KEGG GENES* dataset showed that they are not identical (Supplementary Table 3). In Release 32.0, the associations extracted from the KO dataset are an almost perfect superset of the associations extracted from the *KEGG GENES* dataset. Thus we chose to use the *KO* data set for the KO hierarchy and gene-KO associations and the *KEGG GENES* dataset for the amino acid sequences only.

We implemented a complex regular expression to parse the *KO* dataset and a series of regular expressions to parse the *KEGG GENES* dataset. Parsed information was converted into an intermediate XML format using the Simple API for XML (SAX) and then stored in an embedded relational database using SQLite.

From Release 32.0 we extracted 5540 KO leaf nodes of functional terms, 179 pathways, 748 177 genes (from 19 eukaryotes, 168 bacteria and 19 archaea), and 232 637 documented associations between genes and KO terms.

Automated annotation of new sequences with KO

Given a new DNA or protein sequence, we assign it KO terms based on its similarity to amino acid sequences in the *KEGG GENES* dataset. We calculate similarity using BLASTX for a new DNA sequence and BLASTP for a new protein sequence (Altschul *et al*., 1990, 1997). As choosing the right threshold

Table 2. Most frequent pathways in *P.acnes*, containing more than 20 genes, identified by KOBAS

Pathway	Number of genes
ABC transporters, prokaryotic	128
Ribosome	60
Phosphotransferase system (PTS)	47
Porphyrin andchlorophyll metabolism	46
Oxidative phosphorylation	43
Other replication, recombination and repair factors	40
Purine metabolism	38
Glycerolipid metabolism	37
Glycolysis/gluconeogenesis	34
Pyrimidine metabolism	33
ABC transporters, ABC-2 and other types	32
Other ion-coupled transporters	30
Starch and sucrose metabolism	30
Ubiquinone biosynthesis	28
Glycine, serine and threonine metabolism	28
Pentose phosphate pathway	25
Arginine and proline metabolism	24
Pyruvate metabolism	23
Glutamate metabolism	23
Aminoacyl–tRNA biosynthesis	23
Butanoate metabolism	23
Fructose and mannose metabolism	23
Folate biosynthesis	23
Citrate cycle (TCA cycle)	22
Histidine metabolism	22
Reductive carboxylate cycle (CO_2 fixation)	21

is critical for achieving the most accurate assignments for the most sequences, we tested a variety of different thresholds and combinations of thresholds. In the end we chose as default,

$$E\text{-value} < 10^{-5} \quad \text{AND} \quad rank \leqslant 5,$$

meaning that a new sequence is assigned the KO terms of the first BLAST hit that (1) has known KO assignments, (2) has BLAST E-value $< 10^{-5}$ and (3) has less than five other hits with a lower E-value. The requirement of $rank \leqslant 5$ reduces the possibility of wrong KO assignments and, while it may miss some KO assignments, our evaluation and comparison results (presented in the Results section) indicate that the increase in precision exceeds the loss in sensitivity. These default thresholds can be easily customized by users of KOBAS. Any set of sequences, including whole genomes, can be annotated with the above procedure.

Identification of frequent and enriched pathways

It is often important to identify pathways involved in a set of sequences, such as a cluster of genes from microarray analysis. Since the third level in the KO hierarchy corresponds to KEGG pathways, once we find the right KO terms for a gene, we can trace back through the KO hierarchy to its associated pathways. Given a set of gene (or protein) sequences, we can identify the most frequently occurring pathways. For each pathway that occurs in the set of genes, we count the total number of genes in the set that are involved in the pathway and then rank the pathways by the number of genes.

Since some pathways are larger (i.e. involve more genes) than others, they tend to show up more frequently in any set of genes. Thus it is often valuable to identify the statistically significantly enriched pathways in a set of sequences, given a background distribution. We use the whole genome as the default background distribution (though users can define their own background distribution in the KOBAS system). For each pathway that occurs in the set of genes, we count the total number of genes in the set that are involved in the pathway. We then calculate the p value using a hypergeometric distribution. If a whole genome has N total genes, among which M are involved in the pathway under investigation, and the set of genes has n total genes, among which m are involved in the same pathway, the p value for the pathway is calculated as follows:

$$p = 1 - \sum_{i=0}^{m-1} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}. \tag{1}$$

Because a large number of KEGG pathways are considered, multiple hypotheses tests are performed, which in some cases could result in a high overall Type-1 error (i.e. false positive discoveries) even for a relatively stringent p value cutoff. To reduce the Type-1 errors, we perform an FDR correction (Benjamini and Hochberg, 1995) with default cutoff of 0.05.

KOBAS

We implemented the above methods in KOBAS. It is a standalone command-line program written in Python (2.3.4). It consists of three modules: *kparser*, *blast2ko* and *pathfind*. *Kparser* uses BioPython (1.3.0) and Martel (0.9.0) to parse the *KO* and *KEGG GENES* datasets. The parsed information is managed with SQLite, a small C library that implements a self-contained, embeddable and zero-configuration SQL database engine. *Blast2ko* automatically annotates a set of new sequences (in FASTA format) with KO functional terms. *Pathfind* identifies both the frequent and the enriched pathways in a given set of sequences. It calculates the FDR value by invoking the GeneTS (2.3) (Wichert *et al*., 2004) package of the R (2.00) language (Storey *et al*., 2002) through RPy, an interface from Python to R. KOBAS will run on most Linux systems, and executables are freely available at http://genome.cbi.pku.edu.cn/download.html.

RESULTS

To evaluate the automated annotation of genes using KO as a controlled vocabulary, we first applied KOBAS to two whole genomes, one eukaryotic and one prokaryotic, that have existing KO annotations in KEGG, and compared our automated KO annotations with the existing KO annotations. We then applied KOBAS to a newly sequenced whole genome that has not yet been annotated in KEGG and compared our automated KO annotations with the textual annotations available in the NCBI genome database. To validate the use of KO annotations as an intermediate to identifying metabolic pathways, we applied KOBAS to 14 clusters of genes generated from a microarray experiment and compared our automatically identified pathways with the experimenters' original manual annotations. We also compared annotations of a well-annotated dataset using KO versus GO.

Evaluation of automated KO annotation of the *Saccharomyces cerevisiae* and *Synechocystis* genomes

To evaluate the automated annotation of genes using KO as a controlled vocabulary, we applied KOBAS to annotate the complete sets of genes in a eukaryotic genome, *S.cerevisiae*, and a prokaryotic genome, *Synechocystis* sp. PCC6803. Both genomes are well annotated in KEGG, although not all the genes have been annotated with KO terms, with 1478 (24%) of 6183 total genes in the *S.cerevisiae* genome and 1202 (36%) of 3314 total genes in the *Synechocystis* genome having existing KO annotations. To ensure a rigorous evaluation, we always remove the genome under study from the original KEGG datasets before applying KOBAS.

Table 3. The most frequent and statistically significantly enriched pathways identified by KOBAS for clusters of genes resulting from the microarray experiment

Clusters (annotated by Saldanha *et al.*)	Ten most frequent pathways identified by KOBAS[a]	Statistically Significant (FDR<0.05) pathways identified by KOBAS[b]
3. Stress induced	Starch and sucrose metabolism (10) Butanoate metabolism (9) Glycolysis/gluconeogenesis (8) Fructose and mannose metabolism (7) Galactose metabolism (6) Alanine and aspartate metabolism (5) Glycerolipid metabolism (5) Tetrachloroethenedegradation (4) Bile acid biosynthesis (4) Benzoate degradation viaCoA ligation (4)	Butanoate metabolism (0.003) Fructose and mannose metabolism (0.003) Pentose and glucuronateinter conversions (0.032) Tetra chloroethene degradation (0.032)
6. Leucine induced, sulfate repressed	Valine, leucine and isoleucine biosynthesis (11) Histidine metabolism (7) Phenylalanine, tyrosine and tryptophan biosynthesis (6) Pantothenate and CoA biosynthesis (6) Aminoacyl-tRNA biosynthesis (5) Alanine and aspartate metabolism (4) Urea cycle and metabolism of amino groups (4) Pyruvate metabolism (4) Lysine biosynthesis (3) Other ion-coupled transporters (3)	Valine, leucine and isoleucine biosynthesis (3×10^{-12}) Pantothenate and CoA biosynthesis (2×10^{-6}) Histidine metabolism (4×10^{-5}) Phenylalanine, tyrosine and tryptophan biosynthesis (0.001) C5-Branched dibasic acid metabolism (0.005) Urea cycle and metabolism of amino groups (0.008) Aminoacyl-tRNA biosynthesis (0.018) Alanine and aspartate metabolism (0.035) Other ion-coupled transporters (0.040)
7. Leucine induced	Phenylalanine, tyrosine and tryptophan biosynthesis (3) Cysteine metabolism (3) Nicotinate and nicotinamide metabolism (2) Alanine and aspartate metabolism (2) Histidine metabolism (2) Selenoamino acid metabolism (2) Glycine, serine and threonine metabolism (2) Arginine and proline metabolism (2) Methionine metabolism (2) Urea cycle and metabolism of amino groups (2)	Cysteine metabolism (0.028)
11. Stress repressed, ribosome assembly	Translation factors (17) Purine metabolism (14) Amino acyl–tRNA biosynthesis (10) RNA polymerase (9) One carbon pool byfolate (5) Glutamate metabolism (5) Lysine degradation (5) Glycine, serine and threonine metabolism (5) Ribosome (5) Glyoxylate and dicarboxylate metabolism (4)	Translation factors (2×10^{-6}) Purine metabolism (6×10^{-4}) RNA polymerase (0.003) Aminoacyl–tRNA biosynthesis (0.003)

[a] Numbers in parentheses are the number of genes in the cluster that are part of the pathway.

[b] Numbers in parentheses are the FDR value of the pathway;

We used different evaluation criteria for genes with and without existing KO annotations. For genes with existing KO annotations in KEGG, we used their existing KO annotations as the gold standard for comparison with the automated annotations by KOBAS. We used two measures of annotation quality, precision and coverage, defined as follows:

$$precision = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \qquad (2)$$

$$coverage = \frac{\mathrm{TP}}{N} \qquad (3)$$

where *TP* (*true positive*) is the number of the sequences for which KOBAS' annotation is the same as the original KO annotation, *FP* (*false positive*) is the number of the sequences for which KOBAS' annotation is different from the original KO annotation and *N* is the total number of sequences in the genome that have existing KO annotations. *precision* measures the proportion of sequences with correct KOBAS annotations among all KOBAS annotations, whereas *coverage* measures the proportion of sequences with correct KOBAS annotations among all KEGG-annotated sequences.

There is a tradeoff between high *precision* and high *coverage*. We tested a variety of thresholds, including *identity*, *E*-value and

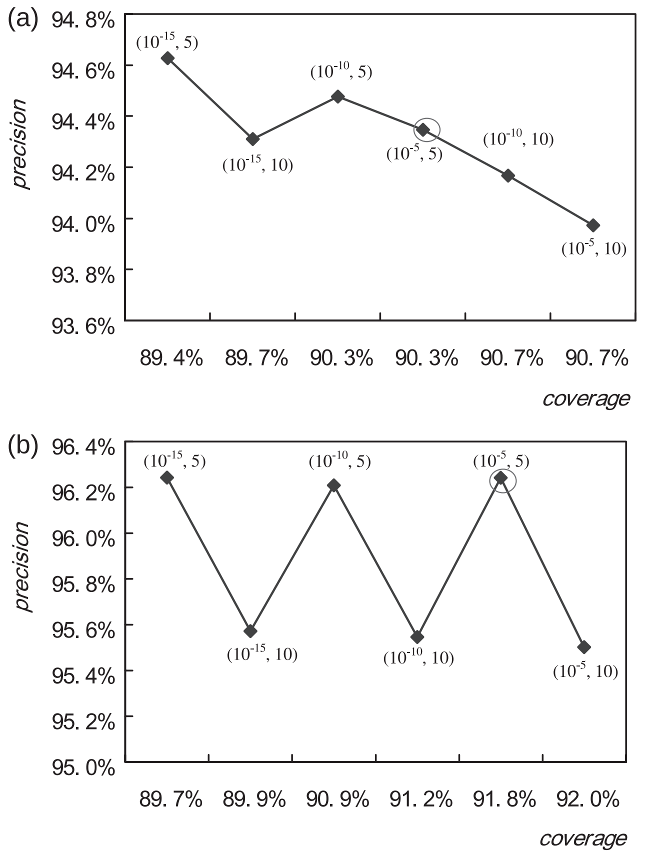

Fig. 1. Evaluation of KOBAS' automated annotation of two genomes using different E-value and *rank* cutoffs. The plots show the coverage (X-axis) and precision (Y-axis) of KOBAS' automated annotation of the *S.cerevisiae* and *Synechocystis* genomes while different (E-value, *rank*) cutoffs are used. The circle in each plot shows the default cutoffs. (a) *S.cerevisiae*, (b) *Synechocystis*.

rank, and found that a combination of E-value and *rank* cutoffs gives the best overall result. Figure 1 shows the *precision* and *coverage* of KOBAS' annotation of the *S.cerevisiae* and *Synechocystis* genomes using different E-value and *rank* cutoffs. With the default threshold, E-value $< 10^{-5}$ and *rank* $\leqslant 5$, the *precision* and *coverage* are respectively 94.3 and 90.3% for *S.cerevisiae* and 96.2 and 91.8% for *Synechocystis*.

For genes that have no existing KO annotations in KEGG, we manually compared KOBAS' annotations to the functional descriptions in other databases such as SGD (the *Saccharomyces* genome database, http://www.yeastgenome.org/). We found that KOBAS is capable of automatically creating meaningful annotations not available in KEGG. Several examples are given in Table 1.

Automated KO annotation of the new *Propionibacterium acnes* genome

We then applied KOBAS to a newly sequenced whole genome that is not yet annotated in KEGG: *P.acnes*, a commensal microbe of the human skin (Bruggemann *et al.*, 2004). KOBAS automatically annotated 1177 (51%) of a total of 2297 genes in the *P.acnes* genome. This percentage is high compared to the percentages of annotated genes versus all genes in the annotated whole genomes currently in KEGG. Figure 2 shows a sample of KOBAS' annotation of *P.acnes*. We manually compared KOBAS' annotations with the functional annotations in the NCBI genome database, and found most of KOBAS' annotations to be correct. In particular, KOBAS was able to automatically identify a number of genes known to be important in *P.acnes*' ability to cope with changing oxygen tension. Examples include oxidative phosphorylation (PPA175-176), F-type H^+-transporting ATPases (PPA1238–1243 and PPA1245), nitrate reductase (PPA507–510), anaerobic dimethyl sulfoxide reductases (PPA515 and PPA517) and glycerol-3-phosphate dehydrogenase (PPA2248–2250). Using the pathway identification function, KOBAS automatically discovered the most frequent pathways in *P.acnes*, as shown in Table 2, including the critical oxidative phosphorylation and nitrogen metabolism pathways. These findings are consistent with what is known about the biology of *P.acnes* (Bruggemann *et al.*, 2004).

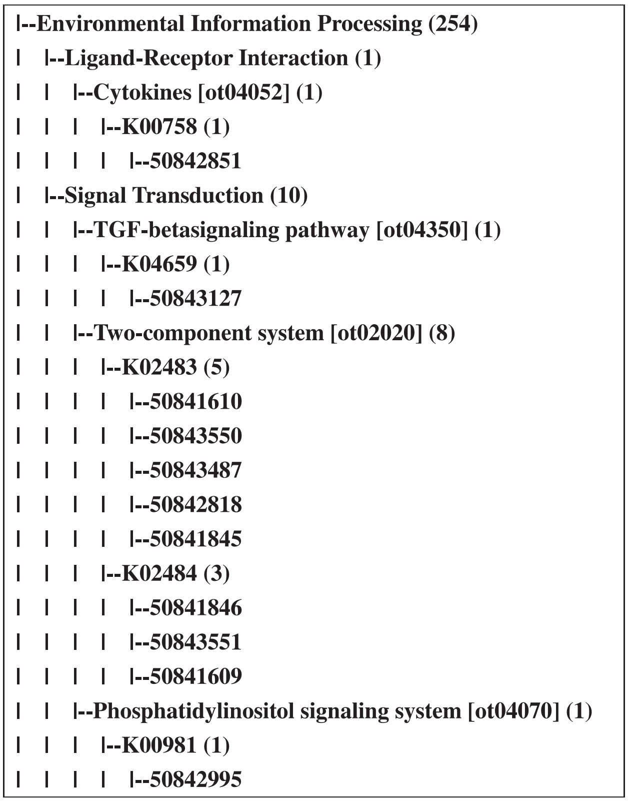
```
|--Environmental Information Processing (254)
|  |--Ligand-Receptor Interaction (1)
|  |  |--Cytokines [ot04052] (1)
|  |  |  |--K00758 (1)
|  |  |  |  |--50842851
|  |--Signal Transduction (10)
|  |  |--TGF-betasignaling pathway [ot04350] (1)
|  |  |  |--K04659 (1)
|  |  |  |  |--50843127
|  |  |--Two-component system [ot02020] (8)
|  |  |  |--K02483 (5)
|  |  |  |  |--50841610
|  |  |  |  |--50843550
|  |  |  |  |--50843487
|  |  |  |  |--50842818
|  |  |  |  |--50841845
|  |  |  |--K02484 (3)
|  |  |  |  |--50841846
|  |  |  |  |--50843551
|  |  |  |  |--50841609
|  |  |--Phosphatidylinositol signaling system [ot04070] (1)
|  |  |  |--K00981 (1)
|  |  |  |  |--50842995
```

Fig. 2. KOBAS' annotation of the newly sequenced *P.acnes* genome. The indentation, shown with '|', denotes a hierarchical annotation of the genome consisting of five levels. The first three levels correspond to the functional categories in the first three levels of KO, and the fourth level corresponds to the KO ID. The numbers in parentheses are the total numbers of genes that are mapped to each functional category or KO ID. The fifth level lists the gene IDs of all the genes that are annotated with each KO ID.

Identification of frequent and enriched pathways in microarray gene clusters

To evaluate KOBAS' ability to automatically identify frequent and enriched pathways in a set of genes, we applied it to the clusters of genes resulting from a recently published microarray experiment (Saldanha *et al*., 2004). In their experiment, Saldanha *et al*. studied the physiological response of *S.cerevisiae* to the limitation of different nutrients in batch and steady-state (chemostat) cultures using a whole-genome microarray. They identified 14 clusters of co-expressed genes and manually annotated the biological relevance of each cluster. For each cluster, we applied KOBAS to first annotate all the genes with KO and to then identify both the most frequent and the statistically significantly enriched pathways. With the rather strict cutoff of FDR < 0.05, KOBAS found statistically significantly enriched pathways in four of the clusters, as shown in Table 3. These automated results are consistent with the authors' annotations of the clusters in the original paper (Saldanha *et al*., 2004).

Comparison of KO and GO annotations on a well-annotated prokaryotic protein set

We compared the KO versus GO annotations of the important photosystems I and II protein family in *Anabaena* (PCC 7120). The dataset consists of 46 proteins, extracted from CyanoBase (http://www.kazusa.or.jp/cyano/) and carefully manually curated. It is available for download at http://genome.cbi.pku.edu.cn/download.html. We annotated the proteins with KO and then with GO. For KO annotations, we used KOBAS with the default parameters, as described in the Methods section. For GO annotations, we used one of the most popular GO annotation approaches which is to BLAST the query sequence against GOA UniProt sequences and select the GO terms of the top hit. KO annotated 44 out of the 46 proteins and all annotations were correct, whereas GO annotated 33 proteins with 28 correct.

DISCUSSION

KOBAS is the first open-access system to use KO as a controlled vocabulary to automatically annotate a set of sequences, such as the complete set of genes in a whole genome or clusters of genes resulting from microarray analysis, and to identify both frequent and significantly enriched pathways. Evaluation results from both whole genomes and microarray gene clusters demonstrate the effectiveness of KOBAS. KOBAS is a standalone application written in Python and is highly modular, making it easy to incorporate into other applications either in part or as a whole.

In transferring KO annotations from known genes to new genes, we used a combination of E-value and *rank* as a threshold rather than E-value alone. Because only one-third to half of the genes in whole genomes currently have existing KO (or GO) annotations, we intentionally generated the BLASTable dataset using the collection of all gene sequences in whole genomes in KEGG, rather than using just those with KO annotations. Thus, for example, should a new gene be found to be similar to 100 top BLAST hits without KO annotations and a 101st hit with KO annotation, we would not assign this KO annotation to the new gene. It is more likely that the new gene is homologous to one of the top 100 genes that have not yet been annotated, rather than the 101^{st} gene that happens to have been annotated. We found that by doing this, even though the total number of KO assignments made by KOBAS is lower, the number of false assignments is also dramatically lower, and we chose to value quality over quantity. Users can easily change this threshold in KOBAS to fit their own analysis goals. In future work we will consider adding more parameters such as percentage of alignment length over the length of the whole genes and concurrence of functional domains, and test their effect on precision and coverage.

A key function of KOBAS is automated identification of pathways in a set of genes or proteins and linking the genes directly to KEGG pathways. KOBAS identifies both the most frequent and the significantly enriched pathways. In determining the latter, KOBAS can use the whole genome, the whole probe set or any user-defined set of genes as the background distribution, and we implemented FDR correction to reduce the number of false positives introduced by multiple hypothesis testing. It is important to note that existing pathway identification systems, such as PathProcessor (Grosu *et al*., 2002), PathMAPA (Pan *et al*., 2003) and ArrayXPath (Chung *et al*., 2004) map genes to pathways based on EC numbers or gene identifiers and thus require that the genes already be in the KEGG pathway database. In contrast, KOBAS uses its automated KO annotation module to identify pathways in sets of genes that are not yet annotated in KEGG, making it especially valuable when analyzing genes in a newly sequenced genome or custom-made cDNA arrays.

Our results indicate that KO is effective as a controlled vocabulary and, although it would be premature to make general conclusions about the power of KO versus GO annotations without further large-scale, in-depth comparisons, some advantages of KO can be observed. At least for some biological systems, such as the important photosystems in the prokaryote genome we tested, KO seemed to provide better annotations in terms of both the total number of proteins annotated and the quality of the annotations. This may be due to the fact that historically KEGG has had a stronger focus on the prokaryotic species than has GO. The biggest limitation of KO is perhaps the fact that it does not yet have as many functional terms in its hierarchy as does GO, although this is improving as KO continues to grow. Overall, we believe that KO-based annotations are complementary to GO-based annotations and systems like KOBAS can make a significant contribution to genomic and proteomic analysis.

CONCLUSION

We developed a KO-Based Annotation System (KOBAS) that can automatically annotate a set of sequences and identify both the most frequent and the statistically significantly enriched pathways. Results from both whole genome and microarray gene cluster annotations with KOBAS are comparable and complementary to existing annotations and demonstrate the use of KO as an alternative controlled vocabulary for automated annotation and pathway identification. KOBAS is a freely available standalone Python program that can contribute significantly to genome annotation and microarray analysis.

ACKNOWLEDGEMENTS

This work was supported by the China National High-tech 863 Program (2004AA231020, 2004BA711A21). We thank Sheng Wei for helpful comments.

REFERENCES

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.

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Ashburner,M. *et al*. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.

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

Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc. B* **57**, 289–300.

Bono,H. *et al*. (1998) Systematic prediction of orthologous units of genes in the complete genomes. *Genome Inform Ser Workshop Genome Inform.*, **9**, 32–40.

Boyle,E.I. *et al*. (2004) GO::TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics*, **20**, 3710–3715.

Bruggemann,H. *et al*. (2004) The complete genome sequence of *Propionibacterium acnes*, a commensal of human skin. *Science*, **305**, 671–673.

Castillo-Davis,C.I. and Hartl,D.L. (2003) GeneMerge—post-genomic analysis, data mining, and hypothesis testing. *Bioinformatics*, **19**, 891–892.

Cheng,J. *et al*. (2004a) NetAffx gene ontology mining tool: a visual approach for microarray data analysis. *Bioinformatics*, **20**, 1462–1463.

Cheng,J. *et al*. (2004b) A knowledge-based clustering algorithm driven by Gene Ontology. *J. Biopharm. Statist.*, **14**, 687–700.

Chung,H.J. *et al*. (2004) ArrayXPath: mapping and visualizing microarray gene-expression data with integrated biological pathway resources using scalable vector graphics. *Nucleic Acids Res.*, **32**, W460–W464.

Dahlquist,K.D. *et al*. (2002) GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nat. Genet.*, **31**, 19–20.

Draghici,S. *et al*. (2003) Onto-Tools, the toolkit of the modern biologist: Onto-Express, Onto-Compare, Onto-Design and Onto-Translate. *Nucleic Acids Res.*, **31**, 3775–3781.

Grosu,P. *et al*. (2002) Pathway processor: a tool for integrating whole-genome expression results into metabolic networks. *Genome Res.*, **12**, 1121–1126.

Hennig,S. *et al*. (2003) Automated gene ontology annotation for anonymous sequence data. *Nucleic Acids Res.*, **31**, 3712–3715.

Kanehisa,M. (1997) A database for post-genome analysis. *Trends Genet.*, **13**, 375–376.

Kanehisa,M. and Goto,S. (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27–30.

Kelso,J. *et al*. (2003) eVOC: a controlled vocabulary for unifying gene expression data. *Genome Res.*, **13**, 1222–1230.

Lee,S.G. *et al*. (2004) A graph-theoretic modeling on GO space for biological interpretation of gene clusters. *Bioinformatics*, **20**, 381–388.

Masseroli,M. *et al*. (2004) GFINDer: Genome Function INtegrated Discoverer through dynamic annotation, statistical analysis, and mining. *Nucleic Acids Res.*, **32**, W293–W300.

Mulder,N.J. *et al*. (2003) The InterPro Database, 2003 brings increased coverage and new features. *Nucleic Acids Res.*, **31**, 315–318.

Pan,D. *et al*. (2003) PathMAPA: a tool for displaying gene expression and performing statistical tests on metabolic pathways at multiple levels for Arabidopsis. *BMC Bioinform.*, **4**, 56.

Saldanha,A.J. *et al*. (2004) Nutritional homeostasis in batch and steady-state culture of yeast. *Mol. Biol. Cell.*, **15**, 4089–4104.

Storey,J.D. (2002) A direct approach to false discovery rates. *J.R. Statist. Soc. B*, **64**, 479–498.

Wichert,S. *et al*. (2004) Identifying periodically expressed transcripts in microarray time series data. *Bioinformatics*, **20**, 5–20.

Young,A. *et al*. (2005) OntologyTraverser: an R package for GO analysis. *Bioinformatics*, **21**, 275–276.

Zeeberg,B.R. *et al*. (2003) GoMiner: a resource for biological interpretation of genomic and proteomic data. *Genome Biol.*, **4**, R28.

Zehetner,G. (2003) OntoBlast function: from sequence similarities directly to potential functional annotations by ontology terms. *Nucleic Acids Res.*, **31**, 3799–3803.