東京大学大学院新領域創成科学研究科
情報生命科学専攻

平成 18 年度

修士論文

A prokaryotic gene annotation tool based on sequence similarity and biomedical text

配列の相同性と生命科学関連文献を利用した

原核生物遺伝子のアノテーション手法の開発

2007 年　3 月提出

指導教員　高木　利久　教授

47-56912　山口　大輔

A prokaryotic gene annotation tool based on sequence similarity and biomedical text

配列の相同性と生命科学関連文献を利用した
原核生物遺伝子のアノテーション手法の開発

Name: Daisuke Yamaguchi

Student ID: 47-56912

Thesis Supervisor: Toshihisa Takagi

Date: 2006-2-1

Abstract

In recent years, a substantial number of prokaryotic genomes were completely sequenced and now additional prokaryotic genome sequencing projects are ongoing. On the other hand, with the development of new sequencer, direct cloning approach without cultivation was made possible and studies of metagenome such as intestinal bacteria florae have been spotlighted. Although many genes have been found in these studies, most of them have not been functionally annotated yet. To address this problem, many computational methods have been proposed such as homology-based annotation transfer, structure prediction, integrative functional genomics, and so on. The most major method for assigning an annotation to a newly sequenced gene is to utilize existing information of its similar sequences. It does not, however, work well in prokaryotic genome sequences, since its similar sequences often have a variety of annotations or lack reliable annotations. Some of similar sequence functions are varied in biomedical literature and are not provided for sequence annotation. The exploitation of biomedical literature is a crucial subject.

Accordingly, we propose a new sequence annotation method and implemented it as an annotation system that can be applied to newly sequenced prokaryotic genes, especially for sequenced ones as metagenome by combining a homology-based technique and biomedical text. In our system, characteristic functional terms of each gene are extracted from document sets including corresponding gene name based on tf*idf and are stored in the database in advance. The similar sequences of the query sequence are searched with BLAST. The characteristic functional terms of similar sequences in the database are assigned as the annotation of the query sequence. More distinctively and more commonly described terms in the document sets of similar sequences are selected in our system to obtain appropriate functional terms.

In this study, to consider various functional words without increasing meaningless words, likely insignificant words were predicted using mutual information between each word and each MeSH term based on their co-occurring frequencies in MEDLINE abstracts and were removed from all words with high tf*idf. In this filtering process, functional words were obtained from all words with F-measure of 57%. When this method was applied to our annotation system, the precision of our annotation system was increased from 18.9% up to 44.6%. This system is expected to be useful for annotations of metagenome sequences.

日本語要旨

近年、数多くの原核生物においてゲノム配列が決定され、現在も複数のゲノムプロジェクトが進行中である。他方、新シークエンシング技術の開発によって微生物の単離・培養を経ず、直接DNAのクローニングが可能となり、その結果メタゲノムのような研究が注目を集めるようになっている。しかし、このような研究から新たに多数の遺伝子が発見されてくる中で、それらの多くは未だにアノテーションできずに残されていることが現在、問題となっている。この問題を解決するために、アノテーション・トランスファーや、構造予測、統合機能ゲノムなど各種の手法が提案されている。これらのうち、機能未知の配列のアノテーションに最もよく用いられる手法は、クエリー配列と相同性をもつ配列の機能情報を活用するというもので、これがアノテーション・トランスファーと呼ばれるものである。しかし、それらの相同配列群が多様なアノテーションを含んでいる場合や、有用なアノテーションが含まれない場合があり、アノテーション・トランスファーが困難であることも多い。相同配列の機能のいくつかは文献中に埋もれており、アノテーションとして簡便に利用することは難しい。従って、生命科学の文献をいかに遺伝子アノテーションに活用できるかが鍵となっている。

そこで我々は、相同性情報と文献情報を利用して、原核生物の新規に決定された配列、特にメタゲノムに対しても適用しうる新たな手法を提案・実装した。本システムでは遺伝子名を含む文献セットから当該遺伝子の機能を示す特徴語を抽出し、tf*idfによってスコア付けを行い、予めデータベースに格納しておく。一方アノテーション対象のクエリー配列については、最初にBLASTを用いて相同配列を収集する。次にデータベースを参照して相同配列に関連付けられた機能を示す用語を取得し、それらをアノテーションとしてクエリー配列に付与する。そこで、適切な機能を示す用語を選択するため、本システムは相同配列群に関する文献セットの中に特徴的に出現し、かつ多くの相同配列に関する用語を選択する。

これらの用語を選択する際には意味のない用語の選択を最小限に抑えて、機能を示す用語を集めなければならない。そこで我々はそれら用語とMeSHタームの出現頻度、及びMEDLINEのアブストラクト中におけるそれらの共起の頻度に基づき、相互情報

量を利用して無意味な用語を予測した。そしてその予測結果を用いて tf*idf の高い用語群から意味のないとされた語を除去し、最終的には機能用語を F-measure で 57%以上で収集することができた。本フィルタリング手法をアノテーションシステムに適用した結果、アノテーションの精度はフィルタリング前後で 18.9%から 44.6%に向上した。本システムは、メタゲノム由来のような配列アノテーションにも有効と思われる。

Table of Contents

List of Tables

Chapter 1

Introduction

Currently, 417 bacteria genomes and 31 archaea genomes have been sequenced and now 1034 bacteria genomes and 59 archaea genomes are being sequenced [8]. In addition to conventional whole genome sequencing, newly emerging field such as metagenomics or environmental genomics is producing a vast amount of sequences. To date, 71 metagenome projects are available to public. Although many genes can be found in these sequences, unfortunately, the most of these genes are not experimentally characterized or annotated yet and such genes are increasing. Because of the large number of these genes, characterization and manual annotation for these genes is not realistic and automated annotation is increasing its importance in this decade [1].

To annotate genes in automatic fashion, heretofore many methods have been proposed and implemented as annotation systems. One of the most popular methods is annotation transfer based on the sequence homology. If a newly determined gene has sequence homology to some well characterized genes, their annotations can be transferred to the newly determined gene. It does not, however, work well since its similar sequences often have a variety of annotations or lack reliable annotations. Manual annotation by curator is sometimes requisite to assign most appropriate annotation to the sequence. Besides annotation transfer based on sequence homology, the integrated method of annotation transfer with some other evidences such as expression profile or protein-protein interaction have been proposed in this decade. This kind of analysis is called "Integrative functional genomics" [6, 16]. For example, a newly discovered gene is linked to genes functionally annotated in other analyses based on protein-protein interactions. Then the commonest function observed among protein interaction pairs would be assigned to the gene [12]. However, it does not work well, too. In spite of the physical interaction, protein partners do not necessarily have the same functions.

On the other hand, though a vast amount of literature about micro-organisms have been published and stored, they have not been leveraged for functional annotations. The challenge of Korbel, J. O. et al [7] is one of few examples that tried to annotate genes of prokaryotes using biomedical literature. They combined phenotypic information in literature and comparative genome analysis. Their method can annotate a gene even if its orthologous group in other organisms has no annotation. However, since their method was designed for the annotation of whole set of genes in a target organism, it is not suitable for the annotation of newly sequenced genes obtained from genomic fragments or metagenomic analysis.

In this study, we propose a new framework for gene annotation and implement it as an annotation system that can be applied to the sequences of metagenomes and environmental genomes. Our system searches similar sequences for a newly discovered sequence using homology search and determines commonly observed feature words among descriptions of these similar sequences. Since our annotation scheme doesn't require other information except similar sequences and articles describing these genes, it can be applied to these sequences.

Chapter 2

Materials and Methods

The schematic representation of our annotation system is shown in Figure 1. The details of each step in Fig.1 and evaluation method of the annotation system is described in this chapter.

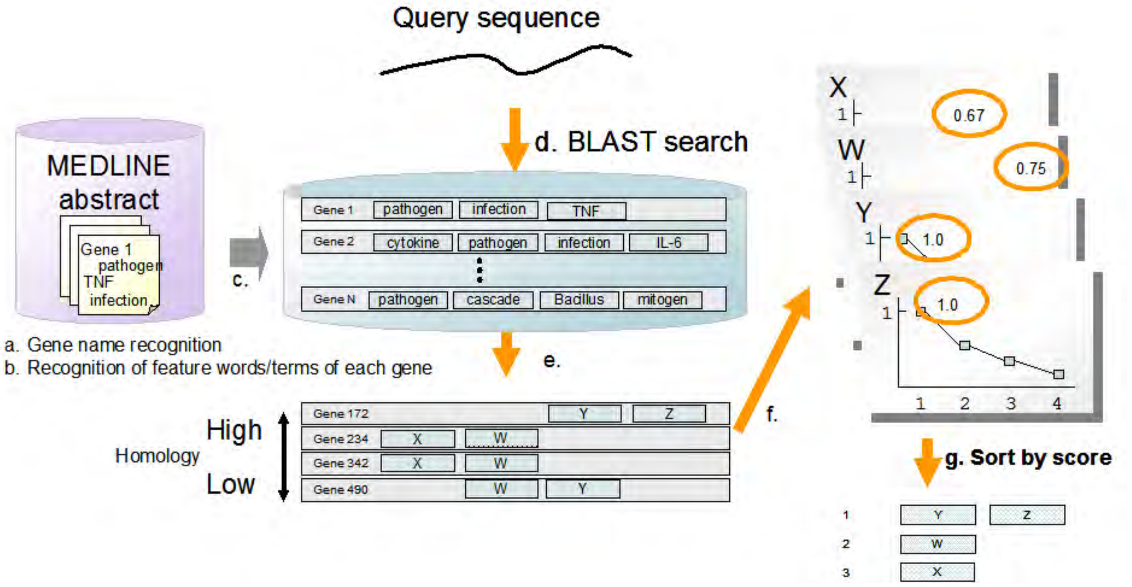

Figure 1. The scheme of our annotation system

The system is composed from two of main part. The first one is indexing subsystem (in figure a, b and c). The index is precompiled by this subsystem and provided to automatic annotation. The second part is automatic annotation subsystem (in figure d, e, f and g).

2.1 Recognition of gene names and selection of feature words/terms

The processes (a)-(c) in Fig.1 are performed in advance. The following explanation of (a)-(g) correspond to the mark in Fig. 1.

(a) Gene name recognition

To utilize gene names described in biomedical literature, we recognized them and created their index as shown in Fig.1 (a).

Since we assumed that prokaryote genes are named systematically than eukaryote ones, we took a dictionary-based approach for gene name recognition in this study. Since most of genes have multiple names (synonyms), we had to scrape together these synonym names as far as possible to judge various gene names as an identical gene description. As far as we know, the largest list of genes and their synonyms is provided as file "gene_info" in EntrezGene [17]. In this list, we regarded symbols, synonyms, locus, full name and description as gene names. We extracted gene names from the list and produced term variations such as conversions between roman numerals and Arabic numerals and replacement of hyphen with white spaces. based on original symbols and aliases. Entrez Gene was containing 850 species and 1,130,886 genes of prokaryotes, and we expanded this dictionary up to 2,103,072 gene names by making variants with this procedure. To search two million of gene names in millions of MEDLINE abstracts, we made a search program using SPARE-Parts that is a library containing Aho-Corasick algorithm. Using Aho-Corasick algorithm we can find gene names in just one scan through MEDLINE abstracts. This search could be completed within several hours in our computer environment (Sun UltraSPARC-IIIi, 1.5GHz). Gene IDs were assigned to detected gene names and gene ID-PubMed ID index was built.

(b) Recognition of feature words/terms of each gene

There are two ideas to obtain feature words of each gene from abstracts. One of the ideas is to use all words appeared in abstracts, and another is to detect terms registered in thesaurus such as GO (gene ontology) or UMLS. We expected these thesauri as non-redundant vocabulary, so we collected terms from UMLS, COGs, TIGR roles and GO as vocabularies for annotation. Currently, the most widely used ontology for gene annotation is GO, but it is originally designed for eukaryote genes, and its applicability to prokaryote ones has not been sufficiently tested yet. This problem is discussed in section 3.2. To search these terms in abstracts, the same program was used for both gene name search and term search. When abstracts were scanned by this program, all of non alphabetical characters were replaced with space.

In addition to these thesauri, we parsed MEDLINE archive formatted in XML and extracted records of MeSH terms attached to each abstract.

(c) Weighting words and terms using tf*idf

After the recognition of words/terms and gene names, abstracts including the corresponding gene name are prepared as document set for each gene. tf*idf values for all recognized words/terms were calculated in every document set to extract feature words/terms of each gene. The *tf* represents "term frequency" , while the *idf* represents "inverse document frequency". When terms are appeared frequently in given documents, they are expected to be significant keywords. On the other hand, when terms are appeared in limited documents, they are expected to be characteristic. This is measured as "document frequency" and its inversed form is *idf*. Thus high tf*idf indicates significance of the word. To consider the effect of document length to tf, we also calculated normalized term frequency factor [13]:

$$tf\,' = \frac{1 + \log(\, tf\,)}{1 + average\ \ (\log(\, tf\,))} \quad \cdots \quad (1)$$

These relations between genes and terms and their tf *idf scores were stored into the index.

2.2 An automated annotation procedure

The automated annotation processes (d)-(g) in Fig. 1 are carried out interactively.

(d) BLAST search

First, using a sequence that a user wants to annotate as the query, he/she does a BLAST search against the database. As a database for BLAST search, we downloaded genomic sequences of prokaryotes from RefSeq [11]. From these sequences we extracted nucleotide sequences using its annotations that is defining the start and end position of the gene in the genome and GeneID. And the amino acid sequences for blastx search were provided by TIGR CMR. The resulting sequence database contains 1.2 million sequences and then they were compiled into indexes for BLAST search. The BLAST search would be done with E-value threshold 0.1 in the default condition.

(e) Acquisition of candidate words and terms for annotation

The result of the BLAST search is a hit-list of similar sequences. The feature words/terms in the document sets of similar sequences (genes) are used for the query sequence annotation in the next (f) step.

(f) Scoring words by commonality in similar sequences

The appropriate feature words/terms are expected to be frequently used in the descriptions of retrieved sequences or specifically used in the descriptions of highly ranked sequences. We developed a method derived from N-best that finds a suitable combination of parameters. In our method, appearance frequencies and ranks in a BLAST hit-list are used for scoring feature words.

In the current implementation, we defined the score for feature words as follows:

$$Score(w) = \max_{1 \le r < N} \frac{freq(w)}{r+1} \quad \cdots \quad (2)$$

where *freq*(*w*) is the frequency of the word contained in similar sequences (hit-list) and *r* means the *r*-th similar sequence. Once the scores are calculated from rank 1 to *N*, finally the highest score is assigned to the word. To normalize the excessiveness of words of top similarity, one is added to the rank. For example, the term “X” is in the second and third of the list of similar sequences in figure 1. Then the scores are calculated for each rank as 0/(1 + 1), 1/(2 + 1), 2/(3 + 1) and 2/(4 + 1), respectively. Finally, the max score 0.5 is assigned to the term “X” when r = 3.

(g) Sorting words by their score

Finally, assigned feature words are ordered by the descending order of the score assigned at previous step.

2.3 Data set for assessment of gene name recognition

To evaluate the performance of gene name recognition in MEDLINE abstracts, we prepared a gold standard from gene2pubmed. We extracted 32,732 of gene-PubMed ID relations as a gold standard from EntrezGene (gene2pubmed) and used it for the calculation of the recall. The data in gene2pubmed, however, contained some relations between genes and articles without any occurrences of corresponding gene names in the abstract. So we used only the abstracts having links to less than or equal to five genes as the gold standard of recall calculation. By this filtering most of relations indicating association between genes in genomic fragments and articles of the genomic fragments were eliminated.

The gene2pubmed was used for obtaining the recall because of its almost perfect precision. But it couldn't be used for the precision due to its extremely low recall. To obtain the precision, therefore, we randomly picked up 100 abstracts from the MEDLINE abstracts in which gene names were recognized and microbe related MeSH terms were attached.

2.4 Data set for assessment of automatically assigned annotation

As far as we know, TIGR CMR is the largest resource of GO-annotated genes. Therefore, the data of TIGR CMR was used as a gold standard for the assessment of annotation performance at first. The resource contained 87 genomes sequenced at TIGR and 268 genomes sequenced at the other organizations. As those files contained GO codes and evidence code, we extract all records having evidence code "TAS (Traceable Author Statement)", "IDA (Inferred from Direct Assay)" or "IMP (Inferred from Mutant Phenotype)". Subsequently, we downloaded sequences from TIGR CMR and extracted corresponding nucleotide sequences to above annotated genes. Thus we chose manually annotated genes, and finally we got 361 genes as the gold standard.

2.5 Selection of feature words/terms for annotation from all of candidate words

There was a plenty of useless candidate words in an obtained result just after the annotation as discussed in section 3.2. To filter out those words, the degree of association between candidate words and MeSH terms were measured by mutual information. In this case, if a candidate word occurs independently from any of MeSH terms, it would give a small mutual-information score, and then it would be a worthless word for annotation.

The mutual information I(x; y) is computed as follows:

$$I(\mathrm{x};\mathrm{y}) = \log_2 \frac{P(x,y)}{P(x)P(y)} \quad \cdots \quad (3)$$

Here, $P(x)$ is the probability of occurrence of the word, and $P(y)$ is the probability of the MeSH term. $P(x, y)$ is the probability of observing x and y together. If there is an association between word x and MeSH term y, then $I(x, y)$ will be much larger than 0. If there was no relationship between x and y, then $I(x, y)$ will be almost equal to 0. If x and y occur in complementary distribution, $I(x, y)$ will be less than 0. Using the mutual information as indicator of cohesiveness, we identified the most closely associated MeSH term for respective candidate words. Thus, if obtained mutual information between candidate word and the closest MeSH term was small, then we can regard the candidate word as worthless.

Here, in conjunction with the mutual information, we consider also about frequencies of candidate words and MeSH terms. Since the terms occurring in low frequencies could be originally valuable, the mutual information was used only for filtering out high frequency words and terms.

In order to determine the best combination of the mutual information and these two frequencies, we used the subset of the gold standard which we made at previous section. The subset was containing annotations of 20 genes, and those were applied to our annotation system. Next, we validated the obtained annotations by hand, and we prepared them for training set. The frequencies were tested at 1, 50, 100, 200, 500, and 1000, respectively. The mutual information was tested from 2.0 to 19 with 0.5 intervals. We tested every combination of these three thresholds on the training set, and we obtained four scores shown below:

A) The precision P_s and the recall R_s of selected valuable words

B) The precision P_e and the recall R_e of eliminated useless words

To consider these obtained four scores all together, we used the following function:

$$E\ measure = 1 - \frac{5+5+1+1}{\frac{5}{R_s} + \frac{5}{R_e} + \frac{1}{P_s} + \frac{1}{P_e}} \quad \cdots \quad (4)$$

This function outputs weighted harmonic mean so that the recalls are considered more importantly than the precisions in order to decrease the improper removal of useful words. Thus the best set of thresholds that minimized the E-measure (maximized the F-measure) was obtained from the training set.

Chapter 3

Results and Discussions

3.1 Evaluation of gene name recognition

As described in 2.3, the gene name recognition performance was investigated by gene2pubmed data and randomly picked up 100 abstracts containing recognized gene names. As a result, the recall was 0.54 and the precision was 0.91. To analyze the reasons of the low recall, we randomly picked up 100 abstracts from false-negatives and checked them. The primary reason was that the corresponding gene names were not described in abstracts but described in tables or main text. The second reason was the deficiency of corresponding gene names in our dictionary, and the third reason was word order variations such as “ribosome recycling factor”, “ribosome releasing factor” or attachment of extra words such as ”p35”, “p35 lipoprotein”. Further extension of our dictionary is necessary to overcome these problems and improve the recall.

Otherwise, it is possible to employ other technique for gene name recognition/extraction. Recently, there have been many techniques and their implementations have been introduced. In the case of prokaryotes, since their genes are named according to the rule of three lower cases and subsequent one upper case, it is quite natural to adopt the technique that can utilize this kind of feature for gene name recognition. For example, Chang and colleagues [4] proposed a technique that exploited these several features of gene names such as lower/upper cases and with/without digits using naive Bayes, maximum entropy and support vector machines as machine learning methods. In their report, their system achieved 83.3% recall and 81.5% precision for genes of human. Although those techniques extract gene names effectively, they can not reliably map extracted gene names to corresponding gene ID [5]. Therefore, if we employ one of those techniques in our system, further improvement is indispensable for associating gene names with sequences.

3.2 Evaluation of extracted terms

We prepared two types of terms for annotation in words/terms extraction procedure. One is gene related feature words extracted from MEDLINE abstracts, and another is terms obtained from existing thesauri (i.e., GO, UMLS, COGs and TIGR roles). These thesauri are non redundant and well structured vocabulary for annotation. However, it is necessary to check whether the contents of these vocabularies are sufficient for annotation or not,

Therefore, we surveyed how much feature words (functional words) were covered by existing ontology/controlled vocabularies. As a preliminary study, we split terms of those vocabularies into words, and checked whether our feature words were covered with these words or not. For the sake of reliability we calculated tf*idf and tf'*idf for each word as described in section 2.1 using the gene-PubMedID link of gene2pubmed. Thus tf*idf scores were separately calculated within their related gene group, subsequently we averaged tf*idf for each words. The list of words was sorted by the averaged tf*idf, and the sorted list was segmented into six blocks of 10,000 words according to the order. Then we picked up 100 words randomly from each block, and checked whether the words were useful for annotation or not and summarized in Figure 2. In the first block about 53% of words are meaningful for annotation and this ratio is decreased according to the rank. From those blocks, we took the first one and checked whether those meaningful words were contained in existing vocabularies and summarized in figure 3. Then 43% of those words were not contained as shown in this figure 3. This result showed that many feature words remained not registered in vocabularies in those thesauri, so we have to develop a method for extending the vocabularies to annotate prokaryote genes properly. These individual words in the graph are listed in Table 1.

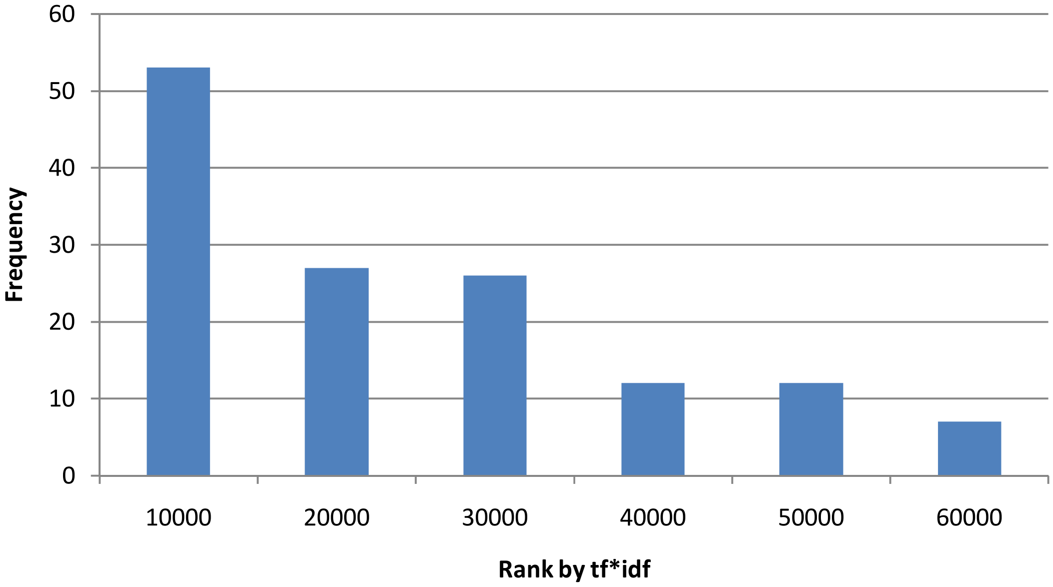

Figure 2. The words covered with COGs/UMLS/GO/TIGR roles.

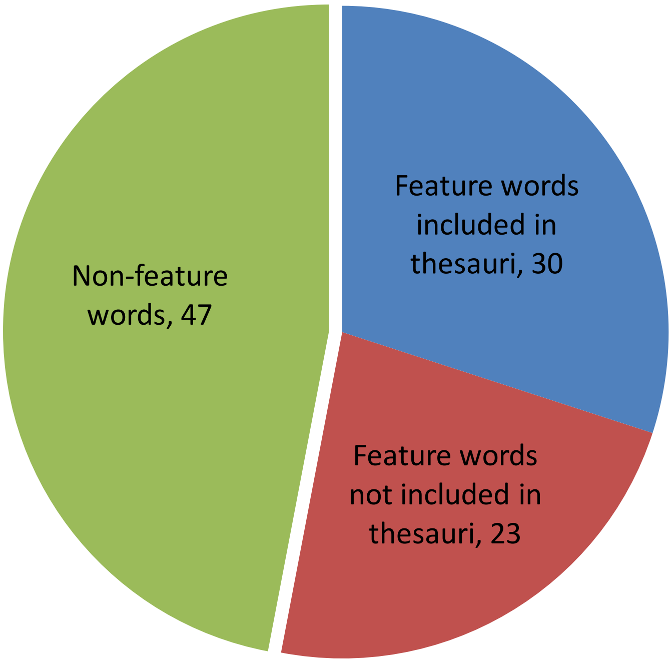

Figure 3. The 100 words of the block of top 10,000

3.3 Evaluation of automatically assigned annotations

When we applied these genes to our system, we do BLAST search using blastn (nucleotides vs. nucleotides), blastx (nucleotides vs. amino acids) and tblastx (nucleotides vs. nucleotides with translating to amino acids). The databases of these sequences were obtained from genome sequences of TIGR CMR as described in 2.2(d). The resulting annotations obtained here were mixture of UMLS terms, COGs classes, TIGR roles, GO terms and individual words. Since these genes were annotated with GO, automatically obtained annotations were compared with these GO annotations, and the result of comparison is summarized as Fig. 4. In this figure apparent difference between blastn and the others were observed. Since blastx and tblastx tended to gather many similar but relatively irrelevant sequences than blastn as expected, irrelevant terms are increased in blastx and tblastx as a results of gathering annotations linked to these sequences. This figure indicates that all three of results have extremely low precision and moderate recall. The precision in Fig 4. is expected quite lower than actual precision, since annotations assigned by GOA were quite limited. Some annotations of them are plausible to be assigned, and thorough investigation of individual annotations is necessary to obtain faithful precision.

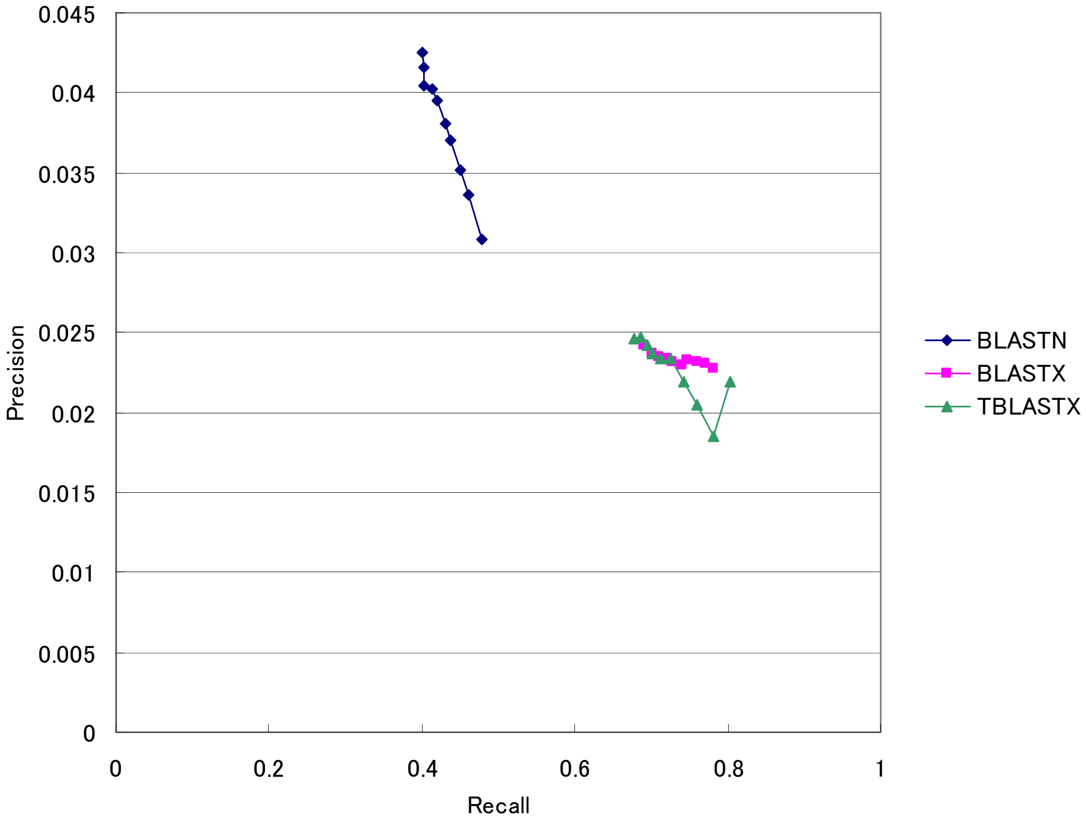

Figure 4. Precision and Recall of GO annotations in our system

Manually annotated data (GOA) were used as a gold standard. Changing E-value threshold from 1.0E-11 to 1.0E-2

Therefore, as a preliminary evaluation, we randomly picked up candidate annotations (words and terms) of 20 genes from gold standard randomly and checked every candidate annotations whether those annotations were appropriate or not. Since these annotations were overwhelming amount to check all of them by hand, we extract every top 100 tf'*idf scored annotations of these 20 genes for manual evaluation. In this evaluation, automatically assigned words were classified into following three classes. If the annotation was apparently correct, then it was classified to “Correct”. If the annotation was apparently false, then it was classified to “Flase”. Otherwise, if we couldn't judge whether the annotation is true or not, then we classified it to “Not Confirmed”. Since the annotations classified to “Not

Confirmed" required further analysis to determine their validity, and it was hard to be done within our limited resources, we reserved them into the class. Finally, there were 2,000 candidate feature words, and 220 of "Correct" words and 151 of "Not Confirmed". From these results the precision obtained here was from 0.110 to 0.189. In the next section of feature words were performed to improve this precision.

3.4 Selection of feature words for annotation from all of candidate words

In the previous section, we showed the recall and the precision of automatically assigned GO annotations, and confirmed the precision of candidate words/terms with manual evaluation. According to these results, the problem was low precision rather than the recall. After we had obtained the list of candidate words, we checked the validity of these words. The example of the list of these words is shown in Table 2. As shown in Table 2, the list contained many English words and common words appeared in biomedical literatures. Although these words indicated high tf'*idf score, they seemed to be not useful for annotating genes. If we can exclude these useless words from the list, we would be able to take only valuable words for annotation.

In general, meaningless terms such as prepositions and demonstrative pronouns are known to be equally distributed overall literature, while meaning/valuable terms including functional terms show biased distribution in literatures. To distinguish these valuable words from meaningless words, the degree of association to MeSH terms was utilized. That is, we assumed that meaningless words would be low/no association to MeSH terms. As a measure of degree of association between MeSH term and a candidate word, mutual information (MI): a measure of mutual dependence between two variables [9], was calculated based on the frequencies of MeSH term and a word and their co-occurrence frequency as shown in section 2.5. MI for all MeSH terms were calculated for a word and the highest MI was adopted to judge whether the word was meaningless word or valuable word. Similar to other statistical measures, MI value is not effective for small numbers of $P(x)$ or $P(y)$. The border value of MI whether meaningless or valuable word is unknown.

Accordingly, we computed mutual information for every combination of three thresholds as described in section 2.5, and we obtained the best set of thresholds that minimized the E-measure. The best performance was achieved F-measure of

0.857 at: Word frequency = 200, MeSH term frequency = 50, and Mutual information = 7 (Figure 5). Then the precision of extracted words was 56.9%.

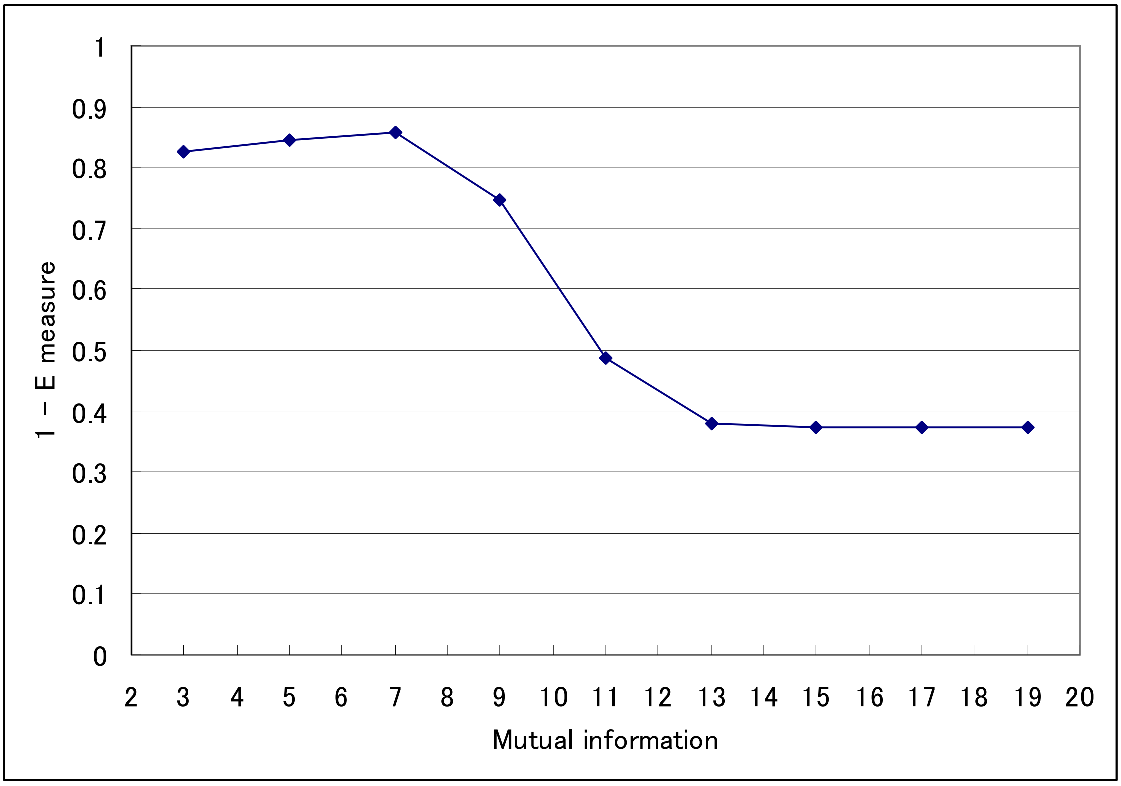

Figure 5. The E measure under the fixed condition of word frequency 200 and MeSH term frequency 50.

We tested every combination of mutual information threshold and frequencies (1, 50, 100, 200 and 500) of candidate words and MeSH terms. The above combination of frequencies gave the maximum F-measure (minimum E-measure) where the mutual information threshold was 7.

3.5 Evaluation of correctness of annotations

Using the thresholds learnt from the training set, we again measured the performance of our system using newly prepared dataset containing randomly picked 30 genes. The subset used here was made not to include the genes contained in the previous subset. We applied these 30 genes to our system, and the resulting annotations were manually validated. For example, Table 3 is a part of annotation. This annotation is a list of words and each word are manually validated. In the list apparently correct annotations are marked as '++' (Correct), and probable annotations are marked as '+' (Probable). Those probable annotations are not confirmed whether each word is correct annotation or not but the terms themselves are meaningful. This measurement resulted in precision from 0.217 to 0.446. Thus, by the filtering words, the precision was improved two times better than 0.189 of section 3.3.

This assessment included the entire words retrieved with BLAST and the index. It is not investigated yet the effect of considering E-value of sequence, or tf*idf in this annotation procedure. To examine these effects, we used the data of 20 genes and 30 genes prepared by the previous section.

For the tf'*idf, tf'*idf threshold was changed from 1 to 10000, and the result precision is shown as Fig. 6. It was extremely low precision before the filtering words (blue ovals and light blue boxes) as described in previous section and they were increased excessively by the filtering (red ovals and yellow boxes). By changing the threshold of tf'*idf, precisions were not so increased except for "Not confirmed"(Correct + Probable) of filtrated. As a result, the increase of precision would be limited even if we set a threshold on tf*idf.

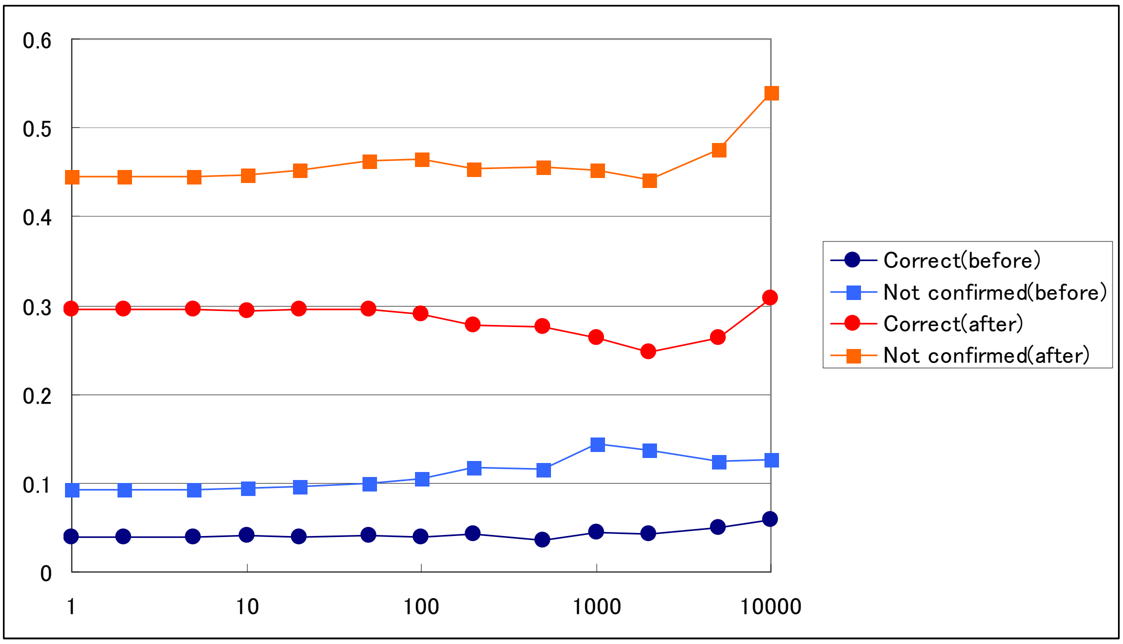

Figure 6. Changes in precision by alternating tf'*idf threshold.

It was extremely low precision before the filtering (blue ovals and light blue boxes), and they were increased excessively by the filtering (red ovals and yellow boxes). Precisions were not so increased except for “Not confirmed” of filtrated.

Since we supposed that precision was also affected by the threshold of similarity in BLAST search, we observed precision with alternating E-value threshold from 1.0E-100 to 0.1. The result of this analysis is represented in Fig. 7. In this analysis, the words shared by multiple similar sequences were regarded as independent annotations. The results showed that no obvious difference was observed even if we changed the threshold of similarity before the filtering. But the differences were emerged by the filtering. Looser threshold than 0.1 will corrupt the precision of annotations. If we set the threshold on E-value, a certain increase in precision will be observed.

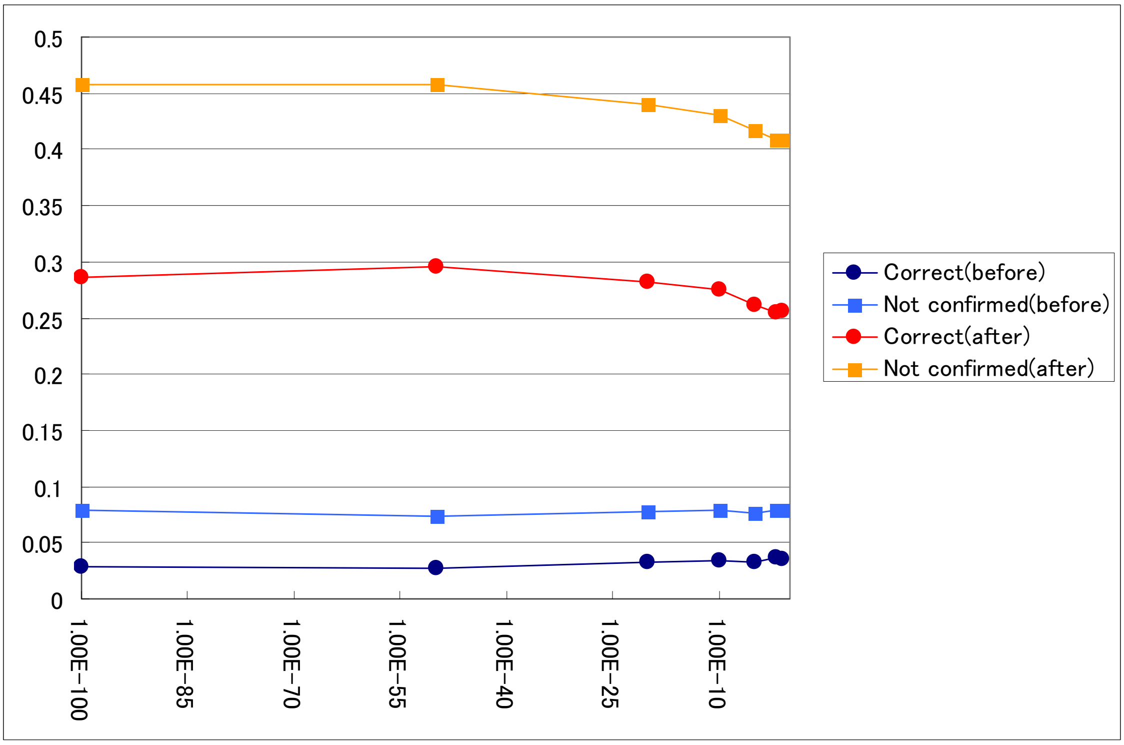

Figure 7. Changes in precision by alternating E-value threshold of blastn.

Before the filtering precisions were extremely low (blue boxes and deep blue ovals). After the filtering precisions were excessively improved (yellow boxes and red ovals). While there were no obvious changes in precisions before the filtering in spite of

E-value threshold change, after the filtering precisions were correlated with E-value threshold.

Chapter 4

Conclusion

In this study, we have developed a new annotation method for prokaryotic genes and implemented it. By combining homology search and biomedical texts, our system enables the annotation of sequences of metagenomic analyses or fragments of genes

There are pros and cons for use of existing thesauri as vocabulary for annotation. In this report, we showed that those vocabularies were insufficient for the annotation of prokaryotic genes. When free words were adopted on the basis of this result, many meaningless words were mixed annotation words. To overcome this problem, we developed the method that filter out those meaningless words from all free words using mutual information as described in section 3.4. Consequently, we could obtain the set of annotations as valuable words and terms with the precision of 21.7-44.6%, thereby it achieved two times better precision than before the word filtering.

However, there remains following subjects to improve our annotation system. For gene name recognition, its performance can be improved by employing more sophisticated technique. Merely an enlargement or expansion of the gene name dictionary would improve the performance immediately. To utilize the commonality and frequency of words in similar sequences, we implemented the scoring system derived from N-best, but we cannot yet utilize it effectively. In this study, we considered only feature words and terms of thesauri for annotation, but phrases or sentence structures are also important for annotation and should be considered together.

The main goal of our study is to bridge the sequence and its functional annotation evidences. It would be also worth integrating some genomic information such as a whole sequence set to our annotation system to provide a certain confidence for annotations made by literature mining.

References

[1] Andrade, MA, Sander, C, Bioinformatics: from genome data to biological knowledge. 1997, Curr. Opin. Biotech. 8, 675-683.

[2] Bodenreider O, The Unified Medical Language System (UMLS): integrating biomedical terminology. 2004, Nucleic Acids Res 32(Database issue).

[3] Bruce WW, Loek C, SPARE Parts: a C++ toolkit for string pattern recognition. 2004, Software: Practice & Experience, 34(7):697-710.

[4] Chang J, Schutze H, Altman R, GAPSCORE: finding gene and protein names one word at a time. 2004, Bioinformatics, 20:216-225.

[5] Jensen LJ, Saric J, Bork P, Literature mining for the biologist: from information retrieval to biological discovery. 2006, Nat Rev Genet. Feb;7(2):119-29.

[6] Karaoz U, Murali TM, Letovsky S, Zheng Y, Ding C, Cantor CR, and Kasif S, Whole-genome annotation by using evidence integration in functional-linkage networks. 2004, PNAS, 101(9): 2888-2893.

[7] Korbel JO, Doerks T, Jensen LJ, Perez-Iratxeta C, Kaczanowski S, Hooper SD, Andrade MA, and Bork P, Systematic Association of Genes to Phenotypes by Genome and Literature Mining. 2005, PLOS Biology., 3(5):e134.

[8] Liolios K, Tavernarakis N, Hugenholtz P, Kyrpides, NC, The Genomes On Line Database (GOLD) v.2: a monitor of genome projects worldwide. 2006, Nucleic Acids Res 34, D332-334.

[9] Manning CD, Schtze H, Foundations of Statistical Natural Language Processing. 1999, MIT Press, 66-68.

[10] Peterson JD, Umayam LA, Dickinson T, Hickey EK, White O, The Comprehensive Microbial Resource. 2001, Nucleic Acids Res. Jan 1;29(1):123-5.

[11] Pruitt KD, Tatusova, T, Maglott DR, NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. 2005, Nucleic Acids Res Jan 1;33(1):D501-D504.

[12] Schwikowski B,* Uetz P, Fields S, A network of protein-protein interactions in yeast. 2000, Nat Biotechnol. Dec;18(12):1257-61.

[13] Singhal, A. and Buckley, C. and Mitra, M, Pivoted Document Length Normalization. 1996, Proc. of SIGIR, 21-29.

[14] Tatusov RL, Fedorova ND, Natale DA, The COG database: an updated version includes eukaryotes. 2003, BMC Bioinformatics. Sep 11;4:41. Epub Sep 11.

[15] The GO Consortium, The Gene Ontology (GO) project in 2006. 2006, Nucleic Acids Res, 34, D322–D326.

[16] Von Mering C, Huynen M, Jaeggi D, Schmidt S, Bork P, Snel B. STRING: a database of predicted functional associations between proteins. 2003, Nucleic Acids Res. Jan 1;31(1):258-61.

[17] Wheeler DL., Yaschenko E, Database resources of the National Center for Biotechnology Information. 2007, Nucleic Acids Res. 35:D5-12.

Table 1. A part of the word list of the first block

Word	Mean	Thesauri	tf*idf
flaviolin	++		114.2
acetyldihydrolipoamide			102.0
minvac		-	85.0
pseudoligand		-	81.5
nonfermenting	++		80.2
nukacin	++		75.2
coumermycin	++		71.5
epoxypimaricin	+		68.0
apophotolyase	++	-	68.0
wzzst		-	66.4
deglycase	++		66.4
dicyanide			64.3
carboxyphosphate	++		62.6
cerein	++		61.6
shufflon	++	-	61.0
ferripyochelin	++		60.8
sirtuins	++		60.5
subcomponent			58.7
aminooxyacetate			58.4
dockerin	++		54.6
dipicolinate	++		53.2
enolpyruvyl	+		52.1
pyocyanin	++		52.0
biosyntheses	++	-	51.5
replicator	++		51.0
archease	++	-	51.0
dimethylnaphthoquinone		-	51.0
pseudobactin	++		50.7
pseudoverdine	++		49.8
phosphoramidase	++	-	48.3
maltooligosyltrehalose	+	-	47.7
acetylcoenzyme	++		45.7
bacteriopheophorbide			45.6
reprint		-	45.6
anticapsin	++		45.4
rhamnan	++		44.8
pyrocatechase	++		43.4
hydroxycobalamin			42.9
sinorhizobial	++	-	40.8
rrinoids	++	-	40.8
endoglycosidase	++		40.4
clinafloxacin			40.1
mevinolin			39.5
lipoyl	++		39.4
swarmer	++		39.2
bacterioplankton			38.6
ransplantation		-	37.5
intraerythrocytic		-	37.2
uveitis			37.0
ceftizoxime			35.8
cosmids			35.8
microcystin	++		34.8
autoprocess	++	-	34.0
pyrrocorphin	++	-	34.0
pluricellular		-	34.0
formycinylhomocysteine		-	34.0
benzyloxycarbonylated		-	34.0
hypercompetence		-	34.0
language			33.6
polyene			33.3
geranylgeranylglyceryl			33.3
polychloroethanes		-	33.2
intercofactor	++	-	33.2
zorbonensis		-	33.2
esterifies		-	33.2
erythronate			33.2
futura		-	33.2
autoassembly		-	33.2
azidodeoxythymidine			32.6
lambdamax		-	32.6
propylthioadenosine	++		32.6
fucitol			32.6
shrinkage		-	32.5
retrons	++	-	32.2
intersubdomain		-	32.2
linolenic	++		32.1
dioxygenolytic	++	-	32.1
filarial			32.1
barrels		-	32.0

++: Meaningful words, +: likely meaningful words, -: Not registered in thesauri

Table 2. An example of automatically assigned feature words for atpG

Rank	Term	GO/TIGR/COG/UMLS	Score
1	atp		1523.7
2	Genes	UMLS:C0017337	823.2
3	ATPase, Aminophospholipid Transporter-Like, Class I, Type 8A, Member 2	UMLS:C1366832	819.8
4	Operon	UMLS:C0029073	628.3
5	atpase		584.1
6	cfo		544.0
7	operon		541.2
8	genes		541.1
9	mrna		489.3
10	subunits		426.5
11	escherichia		425.6
12	ATP synthase	UMLS:C1622485	413.5
13	f		345.8
14	ATP phosphohydrolase	UMLS:C0001473	325.8
15	translational		324.0
16	ATPase	GO:0016887	315.5
17	gene		290.9
18	synthase		285.5
19	Transcription Initiation	UMLS:C1158830	274.8
20	Translation Initiation	UMLS:C1519613	272.6
21	translational initiation	GO:0006413	272.6
22	Escherichia coli	UMLS:C0014834	269.5
23	b		266.9
24	Chloroplasts	UMLS:C0008266	247.6
25	coli		232.9
26	subunit		228.5
27	Proton-Translocating ATPases	UMLS:C0018437	222.9
28	c		218.7
29	Escherichia	UMLS:C0014833	207.5
30	from		206.9

This list is obtained with blastn. The score indicates tf'*idf (cf. 2.1-(c)) and the list is sorted by this score. The rows containing ID in their center column are terms from those vocabularies. There were many words not so well related to the gene atpG (ATP synthase F1, gamma subunit).

Table 3 An example of automatically assigned feature words for glyceraldehyde-3-phosphate dehydrogenase, type I

Rank	Term	Annotation	GO/TIGR/COG/UMLS	Score
1	GO:0048001	++	erythrose 4 phosphate dehydrogenase	5402.858364
2	UMLS:C1151658	++	erythrose-4-phosphate dehydrogenase activity	3705.370959
3	nad			2750.461356
4	glycolytic			2637.503312
5	gallisepticum			2628.865652
6	epd			2357.50846
7	UMLS:C0059572		erythrose 4-phosphate	2065.542722
8	glyceraldehyde			2056.787474
9	3-phosphate			2040.482108
10	UMLS:C0014823		erythrose	1972.060344
11	UMLS:C0029073	+	Operon	1925.048546
12	TIGR_sub1role:116	+	Glycolysis gluconeogenesis	1836.253106
13	gap2	++		1835.729108
14	operon	+		1822.714437
15	cytadherence	+		1739.575894
16	pgk	++		1682.247591
17	UMLS:C1516627		Clinical Research Associate	1573.741981
18	UMLS:C0317807		Mycoplasma gallisepticum	1530.144021
19	UMLS:C0016762	+	Fructosediphosphate Aldolase	1473.866476
20	UMLS:C0017952	+	Glycolysis	1388.8702
21	GO:0006096	+	glycolysis	1388.8702
22	gap1	++		1335.403104
23	UMLS:C0003074		Anion Gap	1314.352611
24	glycolysis	+		1205.26966
25	nadp			1161.72189
26	mgc2			1102.41103
27	UMLS:C0034263		Pyridoxal	1084.063822
28	GO:0000910	+	cytokinesis	1082.368873
29	GO:0007104	+	cytokinesis	1082.368873
30	GO:0009919	+	cytokinesis	1082.368873
31	GO:0016288	+	cytokinesis	1082.368873
32	cra	++		1070.77201
33	cytadhesin	+		1067.681769
34	UMLS:C0034266		Pyridoxal Phosphate	1061.612572
35	UMLS:C0017857	++	Glyceraldehyde-3-Phosphate Dehydrogenases	1042.485459
36	UMLS:C0017534		Giardia	1034.56096
37	UMLS:C0597219		phosphoglycerate	1021.586431
38	nadh			945.973539
39	o157			900.771256
40	phb	++		896.420889