BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 1 2010, pages 22–29
doi:10.1093/bioinformatics/btp613

Genome analysis

Annotation confidence score for genome annotation: a genome comparison approach

Youngik Yang[1], Donald Gilbert[2] and Sun Kim[1,*]

[1]School of Informatics and Computing and [2]Department of Biology, Indiana University, Bloomington, IN 47408, USA

Received on November 17, 2008; revised on October 19, 2009; accepted on October 20, 2009

Advance Access publication October 24, 2009

Associate Editor: Dmitrij Frishman

ABSTRACT

Motivation: The massively parallel sequencing technology can be used by small research labs to generate genome sequences of their research interest. However, annotation of genomes still relies on the manual process, which becomes a serious bottleneck to the high-throughput genome projects. Recently, automatic annotation methods are increasingly more accurate, but there are several issues. One important challenge in using automatic annotation methods is to distinguish annotation quality of ORFs or genes. The availability of such annotation quality of genes can reduce the human labor cost dramatically since manual inspection can focus only on genes with low-annotation quality scores.

Results: In this article, we propose a novel annotation quality or confidence scoring scheme, called Annotation Confidence Score (ACS), using a genome comparison approach. The scoring scheme is computed by combining sequence and textual annotation similarity using a modified version of a logistic curve. The most important feature of the proposed scoring scheme is to generate a score that reflects the excellence in annotation quality of genes by automatically adjusting the number of genomes used to compute the score and their phylogenetic distance. Extensive experiments with bacterial genomes showed that the proposed scoring scheme generated scores for annotation quality according to the quality of annotation regardless of the number of reference genomes and their phylogenetic distance.

Availability: http://microbial.informatics.indiana.edu/acs.

Contact: sumkim2@indiana.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Researchers can sequence genomes of research interest with a fraction of cost using new massively parallel sequencing technology, such as the 454 (Wicker *et al.*, 2006) and Solexa (Illunima Inc., 2007) machines, compared with that by using traditional Sanger sequencing method (Sanger and Coulson, 1975; Sanger *et al.*, 1977). To utilize the high-throughput sequencing technology, there are major informatics research issues to be solved. Among them, cost for genome annotation is a significant hurdle for inexpensive genome projects.

This article proposes a novel genome annotation scoring scheme that can help to reduce the cost for genome annotation significantly, especially for microbial genomes. Although there has been a great progress in genome annotation technology, it is still a common practice that several biologists go through annotation of an open reading frame (ORF) one by one for a period of several months to even a year. This is mainly because there is no automatic genome annotation system that guarantees correctness of genome annotation. There are widely used annotation services such as CMR (http://cmr.jcvi.org/tigr-scripts/CMR/CmrHomePage.cgi) at J. Craig Venter Institute (JCVI) or IMG at Joint Genome Institute (JGI; Markowitz *et al.*, 2008). Annotation of many ORFs from automatic annotation are correct but the main problem is that we do not know which are correct without further assessment. One common problem of automated annotation via homology [e.g. Basic Local Alignment Search Tool (BLAST; Altschul *et al.*, 1990)] is that gene descriptions in databases can be quite variable and different for the same group of homologous genes (Iliopoulos *et al.*, 2003). A more accurate gene description may be found using a consensus of homolog descriptions, including consideration of phylogenetic distance (Mikkelsen *et al.*, 2005), variable quality of the reference genome annotations and consensus analysis of the annotation text. By addressing these aspects of quality in automated annotation, we have developed a novel confidence scoring system for genome annotation, called *Annotation Confidence Score (ACS)*. It works by comparing annotation of a target genome to a set of selected reference genomes. Input to our system is annotation of genes, for example, from widely used annotation services such as CMR at JCVI or IMG at JGI. We believe that the confidence scoring system will be useful in at least two ways:

(1) ACS can be used to 'rank' ORFs in the target genome in terms of their annotation quality. With a cutoff score for ACS that can be determined empirically, annotators can focus only on those with low quality. This will reduce human effort without sacrificing quality of genome annotation. ACS measures quality of annotation successfully, as we show in this article.

(2) ACS can be used for 're-evaluating' genome annotation periodically as new genomes that are close to the target genome become available. Typically, annotation of genome is done in comparison with currently available sequences. New sequences provide valuable information for re-evaluation

*To whom correspondence should be addressed.

of annotations. ACS methods, when used periodically, can highlight which annotations need to be revised, thus improving the quality of genome annotation over time.

2 APPROACH

2.1 Overview of confidence scoring scheme

The score of ACS is a number ranging from 0 to 1.0 to denote how good annotation of an ORF is in comparison with genes in a reference genome set. The input needed to compute ACS of an ORF in a target genome is an annotation of the ORF, the amino acid sequence of the ORF and a set of gene sequences and their annotations in a set of reference genomes. ACS of an ORF is computed by combining sequence similarity and textual (annotation) similarity in comparison with genes in a set of reference genomes. We use three standard sequence similarity methods in the genome context: BLAST (Altschul *et al.*, 1990), bidirectional best hit (BBH) and TRIANGLE (see Section 2.4.1 for a review of these methods). Matches in the reference genome set that are obtained by using sequence similarity matching techniques are weighted using a modified version of a generalized logistic curve (Section 2.3). Textual annotation similarity is measured by cosine similarity of two annotation texts (Section 2.4.2). There are also a number of issues handling annotation texts: removing stop words, stemming (Section 2.6), finding synonyms (Section 2.5) and abbreviation handling (Section 2.5).

Confidence score should convey the important characteristic of a score, a measure to denote excellence as in quality. For example, a score for an exam with a score range from 0 to 100 should convey quality of achievement by the student. A score of 90 should represent a good achievement, while a score of 10 should indicate a poor performance in the exam. We want to design an annotation confidence scoring scheme to represent how good the given annotation of an ORF in a target genome is in comparison with genes in the reference genome set. This is challenging because the number of matches to the gene will vary widely depending on the two characteristics of a reference genome set: the number of genomes in the reference genome set and phylogenetic distance of the reference genomes. The primary design consideration of the confidence score is to make sure that a score reflects excellence of a given annotation by automatically accommodating the characteristics of the reference genome set. The detailed procedure is given in Figure 1.

INPUT: a target genome G_T; reference genomes $\mathcal{G} = \{G_1, \ldots, G_n\}$
OUTPUT: ACS(x) for all $x \in G_T$

Compute all pairwise comparisons of G_i and $G_j : G_i, G_j \in \mathcal{G} \cup G_T$
Compute BBH(G_i,G_j) : $G_i, G_j \in \mathcal{G} \cup G_T$
Compute TRIANGLE(G_i,G_j,G_k) : $G_i, G_j, G_k \in \mathcal{G} \cup G_T$

for each ORF $x \in G_T$
  ACS_BLAST(x) = COMPUTE_ACS(x, BLAST);
  ACS_BBH(x) = COMPUTE_ACS(x, BBH);
  ACS_TRIANGLE(x) = COMPUTE_ACS(x, TRIANGLE);
  if (count(TRIANGLE(x)) $\geq \theta_{TRIANGLE}$)
  ACS(x) = ACS_TRIANGLE(x)
  else if (count(BBH(x)) $\geq \theta_{BBH}$)
  ACS(x) = ACS_BBH(x)
  else
  ACS(x) = MAX(ACS_BLAST(x),ACS_BBH(x),ACS_TRIANGLE(x))
end for

COMPUTE_ACS(x, s)
  if (s == BLAST) $\mathcal{M}$ = MATCH_BLAST($x, \mathcal{G}$);
  if (s == BBH) $\mathcal{M}$ = MATCH_BBH($x, \mathcal{G}$);
  if (s == TRIANGLE) $\mathcal{M}$ = MATCH_TRIANGLE($x, \mathcal{G}$);

  Replace synonyms in annotation, ANNO(x), of x;
  Remove stop words and handle stemming in ANNO(x);
  for each $m \in \mathcal{M}$
    Replace synonyms in ANNO(m);
    Remove stop words and handle stemming in ANNO(m);
    Compute cosine similarity of ANNO(x) and ANNO(m);
  endfor

  if (s == BLAST) compute ACS using equation 6;
  if (s == BBH) compute ACS using equation 3;
  if (s == TRIANGLE) compute ACS using equation 3;
  return ACS;
END_OF_COMPUTE_ACS

Fig. 1. Overview of an algorithm to compute ACS. Two parameters to the ACS computation, θ_{TRIANGLE} and θ_{BBH}, were set to median values, respectively.

2.2 Design consideration for confidence score

A confidence score of an ORF x, ACS(x) is computed in two steps: (i) collect $\mathcal{M}$ matches in a set of reference genomes to x and (ii) compare annotations of x and genes in $\mathcal{M}$ to compute ACS(x). There are three major design considerations for ACS:

(1) *Adjustment for the number of reference genomes:* a scoring scheme that simply counts the number of matches to the ORF can produce a score that takes a value that simply reflects the number of matches to the ORF that depends largely on the number of reference genomes. Thus, the score that counts the number of matches would fail to reflect how good the current annotation of the ORF is. To be a score that reflects excellence as in quality, ACS needs to automatically adjust the number of expected matches in a given reference genome set. We achieved this goal by using a modified version of a generalized logistic function (Section 2.3) and by adjusting two parameters of the logistic function 'automatically' to reflect the characteristics of the reference genomes set (see Section 2.3 for the two parameters).

(2) *Adjustment for phylogenetic distance of reference genomes to the target genome:* in general, phylogenetically close genomes share more genes than distant genomes. Thus, it is necessary to adjust the number of matches according to phylogenetic distance. This is also achieved by adjusting two parameters of the logistic function 'automatically' to reflect the expected number of matches considering phylogenetic distance. See Section 2.3 for the two parameters.

(3) *Handling in consistency in textual annotation:* once matches to the target gene are selected by sequence similarity, the next step is to compare textual annotation information by using cosine similarity (Section 2.4.2). Comparing free texts [as opposed to controlled vocabulary terms such as Gene Ontology (GO)] needs to handle a number of issues such as removing stop words, stemming (Section 2.6), and handling

synonyms and abbreviations (Section 2.5). To handle synonyms and abbreviations, we used BioThesaurus (Liu *et al.*, 2006a, b) and Medical Subject Headings (MeSH; NIH, 2007) databases (Section 3.4).

2.3 Growth functions for expected number of matches

As we discussed in Section 2.1, the number of matches to an ORF x will vary significantly depending on the number of reference genomes and phylogenetic distance of reference genomes, so does ACS(x). To make ACS a real score that reflects annotation quality consistently regardless of the number of reference genomes and their phylogenetic distance, we explored many functions that could be used to adjust the expected number of matches in a set of reference genomes and found that a revised version of the generalized logistic curve worked well.

Generalized logistic curve, also known as Richards' curve (Centre for Horticulture and Lanscape; Richards, 1959), is a flexible growth function that has been widely used in biology, especially in botany. We found that this function empirically fits data well for the expected number of matches. Since our purpose of using growth function is to weight the number of matches of an ORF x in N_T reference genomes properly rather than estimating the growth rate of a plant, we simplified the generalized logistic curve by using only two parameters to handle the shape of curve: l (lower asymptote) and m (the time of maximum growth) by fixing the other parameters to one. These two parameters are automatically adjusted for a given reference genome set. We first compute a distribution of the number of matches defined by each of the three sequence similarity methods: BLAST, BBH and TRIANGLE and collect 1st quartile and 3rd quartile of the distribution. Then l is set to $1-(\text{3rd quartile}/N_T)$ and m is set to the 1st quartile. This fits very well to a number of reference genome sets (Section 3.1).

$$f(x)=l+\frac{1-l}{1+e^{-(x-m)}} \tag{1}$$

where l = lower asymptote and m = the time of maximum growth.

2.4 Scoring scheme

ACS is computed by using two similarity measures: sequence similarity and annotation similarity. In this section, we review the two similarity measures briefly.

2.4.1 Sequence similarity Given an ORF x in a target genome, matches to x are collected using three standard sequence similarity measures: BLAST, BBH and TRIANGLE. The BLAST, the most widely used sequence alignment method, finds regions of local similarity between two sequences (Altschul *et al.*, 1990). Any matches with e-value less than a given cutoff score are matches to x. BBH is also a very widely used method that selects the bidirectional best hits in the whole genome context. Two genes x_a in G_A and x_b in G_B are BBHs if and only if x_a as a query sequence is best matched to x_b and x_b as the query sequence is best matched to x_a. BBH is widely used to whole genome analysis, e.g. for metabolic pathway analysis (Overbeek *et al.*, 1999) and for gene function characterization (Tatusov *et al.*, 1997). The TRIANGLE method extends BBH to three genes in three genomes, to improve the confidence in accurate identification of orthologs. Three genes, x_a in G_A, x_b in G_B and x_c in G_C, form a TRIANGLE if and only if x_a and x_b are BBHs, x_b and x_c are BBHs and x_c and x_a are BBHs.

2.4.2 Annotation similarity To compare textual annotation similarity, we used cosine similarity because it is a simple and effective measure of text similarity. To compute cosine similarity between two annotations, both annotations are transformed as vectors of words. Then, cosine of two vectors is used as a similarity of two annotations. Cosine of two vectors is defined as below.

$$\cos(\vec{t_1},\vec{t_2})=\frac{\vec{t_1}\cdot\vec{t_2}}{\|\vec{t_1}\|\|\vec{t_2}\|} \tag{2}$$

Supplementary Material 1 shows an example of computing a cosine similarity of two annotations, '*Glutathione homocystine transhydrogenase*' and '*Glutathione CoA glutathione transhydrogenase*'.

Now we resolved how to handle matches by sequence similarity and how to compute textual annotation similarity and we will proceed to show how to combine both sequence and annotation similarity information into a single metric. For each ORF x, ACS(x) is defined as a level-wise best score scheme using each of three sequence similarity methods: $\text{ACS}_{\text{BLAST}}(x)$ for BLAST, $\text{ACS}_{\text{BBH}}(x)$ for BBH and $\text{ACS}_{\text{TRIANGLE}}(x)$ for TRIANGLE.

2.4.3 Confidence score for BBH and TRIANGLE Each of BBH and TRIANGLE methods defines matches to an ORF x in a set of reference genomes. How many genomes have matches to x is termed as *support*, borrowing a concept from the frequent pattern mining problem in the data mining community. As we discussed in Section 2.2, ACS(x) should dynamically adjust the number of expected matches to x for a given reference genome set. This is done by adjusting two parameters of ACS, l and m, by using a distribution of support values. In addition, biologists may want to weight each genome differently. For example, setting w_{G_i} appropriately, a homolog found in a close genome can be weighted more than that in a remote genome, or a homolog found in well-studied genome can be weighted more than that in a genome in poor quality (Section 3.6). This can be done by specifying a genome weight, w_{G_i} in the questions for ACS. The formula to compute ACS for BBH and TRIANGLE is in Equation (3). In the first term of equation, cosine similarities between an ORF and homologs are weighted by the number of supports and then normalized by the sum of all weights. A genome weight w_{G_i} is applied to cosine similarity between x and g_{G_i} to differentiate the effect of the matching gene in the confidence score. For example, setting w_{G_i} appropriately, a homolog found in a close genome pair can be weighted more than that in remote genome, or a homolog found in well-studied genome can be weighted more than that in a genome in poor quality (Section 3.6). The second term of the equation is a generalized logistic regression curve where two important parameters of the logistic function are determined automatically by support rank: lower asymptote l by $1-(\text{3rd quartile}/N_T)$ of support and the time of maximum growth m by 1st quartile of support.

$$c(x)=\left\{\frac{\sum_{i=1}^{N}\cos(\overrightarrow{x},\overrightarrow{g_{G_i}})\times w_{G_i}}{\sum_{i=1}^{N}w_{G_i}}\right\}\times\left\{l+\frac{1-l}{1+e^{-(N-m)}}\right\} \tag{3}$$

where

x = target ORF

$c(x)$ = confidence score of x
$\overrightarrow{x}$ = vector representation of annotation of x
g_{G_i} = homolog found in reference genome $G_i : 0 \leq i \leq N$
N = the number of genomes that have matches
$\overrightarrow{g_{G_i}}$ = vector representation of annotation of g_{G_i}
w_{G_i} = a weight of genome $G_i : w_{G_i} > 0$
l = lower asymptote
= 1 − (3rd quartile/N_T)
N_T = total number of reference genomes
m = the time of maximum growth
= 1st quartile

2.4.4 Confidence score for BLAST While computing ACS_{BBH} and $\text{ACS}_{\text{TRIANGLE}}$, all matches in the genome are treated equally unless the genome weight w_{G_i} is applied. This is because both BHH and TRIANGLE use additional constraints, genome-wide best hits, to define matches to an ORF so that matches are usually of high specificity (correctness), sacrificing sensitivity. However, ACS for BLAST matches, $\text{ACS}_{\text{BLAST}}$, should consider how good the matches are and should treat matches differently according to their match strength to the query. There are several choices for incorporating the match significance. We use ranks of matches to a given query sequence, which is the default option for computing $\text{ACS}_{\text{BLAST}}$. In the rank-based system, a weight for a matching gene is assigned as $N - r + 1$, where N is total number of hits and r is the rank of the gene. Alternatively, $-\log(e\text{-value})$ and bit score can be used for weight scheme. To use $-\log(e\text{-value})$, we needed to set a lower bound of e-value to avoid allowing e-value of 0. We set it as $1.0e-180$. In our experiment, it did not show significant differences among rank, $-\log(e\text{-value})$, and bit score system (Section 3.5).

The equation for computing $\text{ACS}_{\text{BLAST}}$ is:

$$c(x) = \left\{ \frac{\sum_{i=1}^{N} \cos(\overrightarrow{x}, \overrightarrow{g_{G_i}}) \times r'(g_{G_i})}{\sum_{i=1}^{N} r'(g_{Gi})} \right\} \times \left\{ l + \frac{1-l}{1+e^{-(N-m)}} \right\} \quad (4)$$

where

$r'(g_{G_i})$ = reversed rank of BLAST match g_{G_i}
= $N - \text{rank}(g_{G_i}) + 1$

2.5 Synonyms and abbreviation handling

Synonyms and abbreviations used in gene annotation make annotation of the same function syntactically different. In particular, function descriptions of a gene can be different at different sources (e.g. biological databases). Some annotations use abbreviations whereas others use full terms or different abbreviations. In this case, annotations of homolog matches in a reference genome are significantly different in terms of syntax (words used) even if the annotations have a same meaning semantically. A cosine textual similarity of two semantically identical annotations could be low due to their syntactic difference. To consider the semantic similarity of annotations, it is desirable to collect all the variants of gene names and annotations, and cross-references from many sources. Then one needs to compare all variant forms of annotation between target ORF and their homologs. This is a fundamental problem in dealing with annotation, and there are several databases that compile words that describe the same biological function.

To compute ACS, we used BioThesaurus, a web-based system designed to map a comprehensive collection of protein and gene names to UniProt Knowledgebase (UniProtKB) protein entries (Liu *et al.*, 2006a, b). It covers all UniProtKB protein entries, and consists of several millions of names extracted from multiple resources based on database cross-references in iProClass (Huang *et al.*, 2003). Utilizing BioThesaurus, we collected variants of annotation and names of homolog matches.

In addition, protein function is often described using abbreviations and other nomenclature references such as EC number. To handle these issues, we use MeSH (NIH, 2007). MeSH is National Library of Medicine's controlled vocabulary thesaurus and has been used for indexing articles for the MEDLINE/PubMED database. We used three record types, descriptors, qualifiers and supplementary concept records, to handle abbreviations and other nomenclature referencing. MeSH is often complementary to BioThesaurus in a sense that synonyms that are not found in BioThesaurus are found in the MeSH database. Use of GO rather than annotation in free text would be attractive. However, newly sequenced genomes rarely have curated GO controlled vocabulary. That is why we developed ACS using free text annotations that can be used in more general settings.

2.6 Stemming

Words used in annotation may have morphological variants, such as 'transportation' and 'transporting'. Those words will be considered as separate words although their meanings are the same in terms of semantics. ACS should consider these variations of a word. Stemming is a device to help match a query term with morphological variant in the corpus (Chakrabati, 2002). Stemming finds a root of a word and replaces the word with the root. There are two popular approaches of stemming: Porter's algorithm (Porter, 1980) and WordNet (Christiane, 1998). Porter's algorithm replaces a word with the root of the word by considering variants of suffixes of a word. On the contrary, the WordNet approach utilizes a database to look for a root of a word and replaces the word with its root. Since WordNet uses a curated database of word roots, the WordNet approach is more accurate than Porter's approach in general. Thus, we used the WordNet approach for ACS computation.

3 RESULT

We performed a series of experiments to evaluate ACS using UniProtKB/Swiss-Prot as a main source of sequences and annotations for target and reference genomes. In Section 3.1, we tested how well distributions of the number of matches in a reference genome set fit logistic curves on various settings of reference genome set. In Section 3.2, we demonstrated that ACS is a score representing the quality of annotation with four different target genomes in terms of the issues discussed in Section 2.2. In Section 3.3, we examined the ACS distribution of genes with randomly replaced annotations to present some guideline for ACS cutoff setting. Section 3.4 showed variation of ACSs after handling textual variations. In Section 3.5, we compared the rank-based ACS for BLAST to the alternative ACS scoring schemes. Section 3.6 showed the experimental results using various weighting criteria.

3.1 Logistic curve adjustment

To test how well distributions of the number of matches in a reference genome set fit logistic curves on various settings of reference genome set, we performed extensive experiments to see

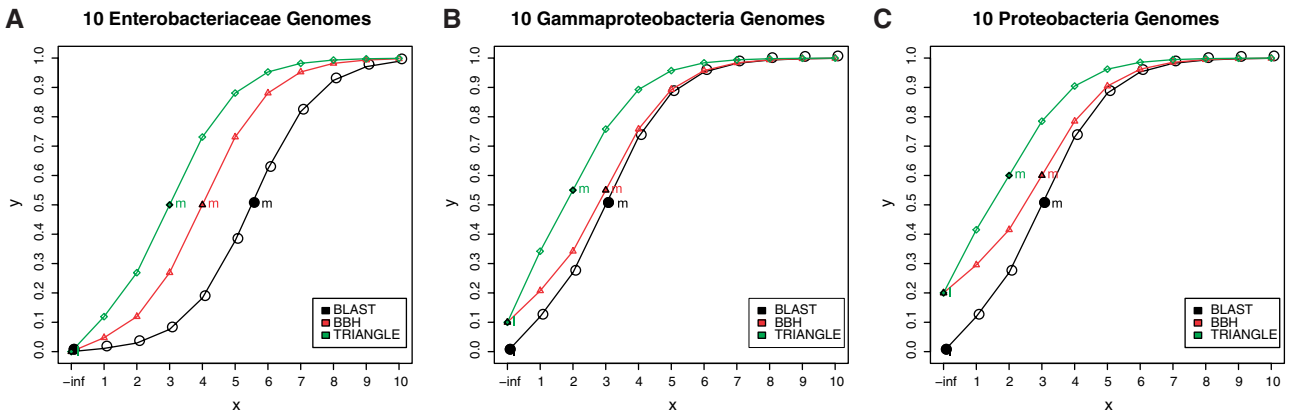

Fig. 2. Distribution of the number of matches in reference genomes of various phylogenetic distances. The label *x* is the number of matches to the query sequence and the label *y* is the ACS value computed for the query. The data point *l* is the lower bound of the logistic curve and the data point *m* is the value where the logistic curve reaches at the maximum growth. As shown, the plots fit well the logistic curves. *Yersinia enterocolitica subsp. enterocolitica str. 8081 (YERE8)* was used as a target genome. Ten genomes were randomly selected as reference genomes from *Enterobacteriaceae* (**A**), *Gammaproteobacteria* (**B**) and *Proteobacteria* (**C**), respectively, in the order of phylogenetic distance to the reference genomes. In each of the plots, three curves show the match distribution using BLAST, BBH and TRIANGLE. As the reference genome set was more distant (A–C), the shape of the curve increased fast and the lower bound of the curve also increased. Note that two parameters in the logistic curve equation can effectively model the change in the curve shapes.

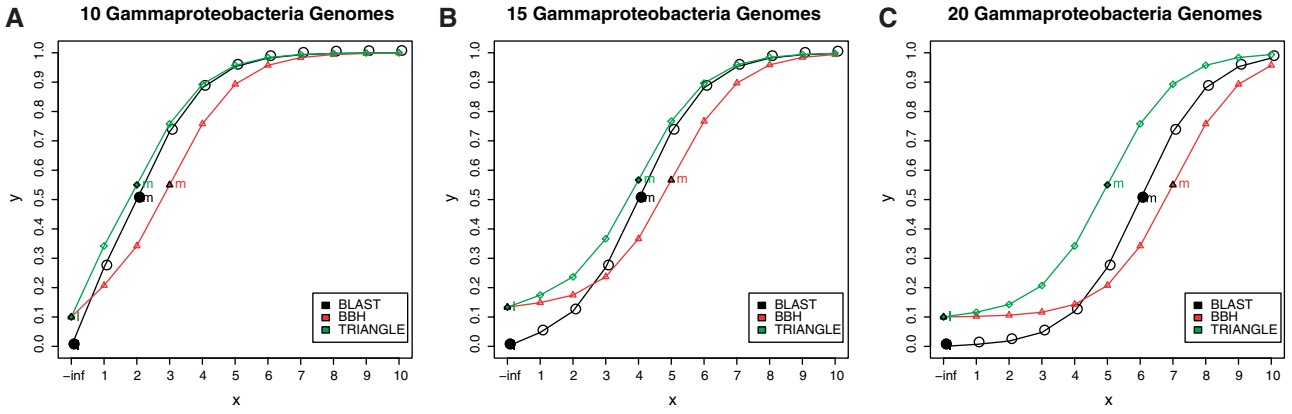

Fig. 3. Distribution of the number of matches in reference genome sets with different number of genomes: 10 (**A**), 15 (**B**) and 20 (**C**). The label *x* is the number of matches to the query sequence and the label *y* is the ACS value computed for the query. The data point *l* is the lower bound of the logistic curve and the data point *m* is the value where the logistic curve reaches at the maximum growth. As shown, the plots fit well the logistic curves. *Yersinia enterocolitica subsp. enterocolitica str. 8081 (YERE8)* was used as a target genome. In each of the plots, three curves shows the match distributions using BLAST, BBH and TRIANGLE. As fewer genomes were used, the curve grew fast and the lower bound of the curve also increased. Note that two parameters in the logistic curve equation can effectively model the change in the curve shapes.

the effect of the number of reference genome sets and the effect of phylogenetic distance using three sequence matching techniques BLAST, BBH and TRIANGLE. Figures 2 and 3 are example of how a modified logistic curve fits with various genome selections. In all the experiments, the logistic curve fit very well the distribution of the number of matches.

3.2 Effect of the number of reference genomes and their phylogenetic distance on ACS

In the previous section, we showed that the number of expected matches fits well the logistic curve for different number of reference genomes and for reference genomes with various phylogenetic distances. Here, we demonstrated that ACS is a score reflecting the quality of genome annotation in various situations with different number of reference genomes and with reference genomes of various phylogenetic distances. Supplementary Material 2 shows that ACS of the same gene does not change much by the number of reference genomes and their phylogenetic distance.

3.3 ACS cutoff setting

It is difficult to advise what value the ACS cutoff sets to. Setting a cutoff value for ACS is analogous to setting an *e*-value cutoff for BLAST searches, on which no consensus has been made yet. However, it will be useful to have some guideline for ACS cutoff setting. Thus, we performed a series of experiments with well-annotated genomes. In the experiment shown in Figure 4, we used

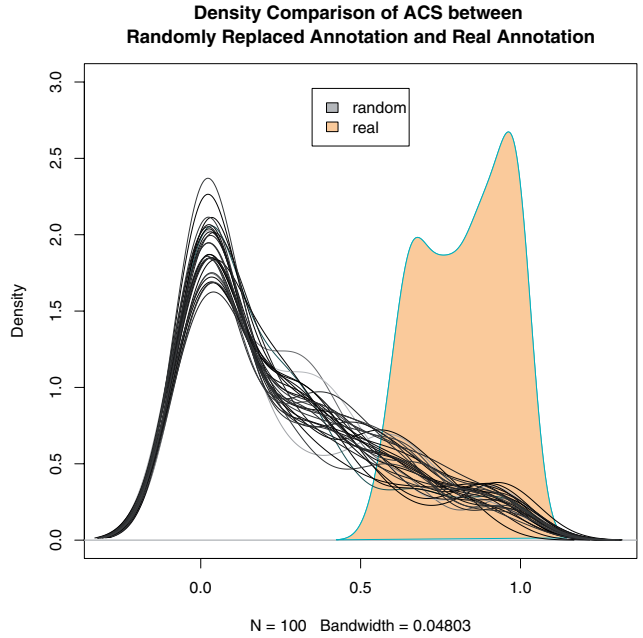

Fig. 4. ACS distribution comparison between original annotations and annotations of 100 randomly sampled genes from *E.coli*. The original annotation of each gene was replaced by the annotation of a randomly selected BLAST match and ACS of the gene was recalculated. For each gene, this process repeated 30 times. The filled density plot with a label 'real' on the right side is the density of ACSs of 100 original annotations. Each of unfilled 30 density plots with label 'random' on the left side is a density plot of ACSs of the 100 genes with randomly replaced annotations. At ACS value of 0.5, overlaps between the density plots were minimal, which suggested that 0.5 would be a good ACS cutoff value.

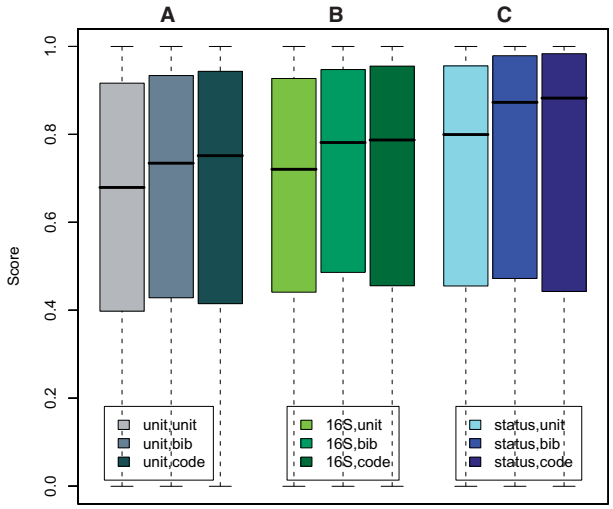

Fig. 5. Confidence score with nine combinations of genome and gene weights in three panels: (A) uniform, (B) 16S phylogeny, and (C) status. The legend "xxx,yyy" represents a genome weight 'xxx" and a gene weight "yyy." Box plots for three genome level weights are grouped with three gene level weights. See detail in text. As shown, both genome and gene weights improved ACS in terms of the median ACS and the range of ACS as an indication of a better discrimination.

100 genes randomly sampled from *E.coli*. These 100 genes were manually inspected so that only meaningful annotations were used. For example, genes with annotation something like 'uncharacterized protein' were not included. Fifteen *Gammaproteobacteria* genomes were used as reference genomes. Then, original annotation of each gene was replaced by annotation of a randomly selected BLAST match and ACS of the gene was recalculated. For each gene, this process repeated 30 times. The results of ACSs of 100 genes were summarized as density plots in Figure 4 and box plots in Supplementary Material 5. The density plots were generated using `density` function with `gaussian` kernel in R. The filled density plot with a label 'real' on the right side in Figure 4 is the density of ACSs of 100 original annotations. Each of unfilled 30 density plots with a label 'random' on the left side is a density plot of ACSs of 100 genes with randomly replaced annotations. As shown, there were only small overlaps between ACS density plots with original annotations and with randomly replaced annotations; note that some of randomly replaced annotations can be the correct ones. The difference between ACS distributions is also clear in the box plots (Supplementary Material 5). Both the density plots and the box plots suggested that ACS of gene with a randomly replaced annotation be unlikely to go beyond 0.5. In addition, the choice of the ACS cutoff value of 0.5 is reasonable since about 60–70% of genes in many genomes in experiments with many different setting are >0.5 (for an example, see Fig. 5), thus annotators will need to look at about 30–40% of genes in a target genome manually. Although 30–40% of genes are many, there are quite a number of genes with not very meaningful annotations such as 'hypothetical proteins' or 'uncharacterized proteins', which annotators would not spend much time on.

3.4 Textual variations

We performed experiments with *Escherichia coli K12* as a target genome and 10 reference genomes in *Enterobacteriaceae* to study the effect of text comparison issues on ACS. Supplementary Material 3 shows examples of the improvement of reliability in ACS after removing stop words and by using MeSH lookup.

3.5 Performance of the rank-based ACS for BLAST

By default, we use a rank-based approach for computing ACS for weighting BLAST matches. Supplementary Material 4 shows that there is no significant difference among the three scoring schemes: the rank-based, $-\log(e\text{-value})$ and bit score.

3.6 Effect of different weight schemes for ACS

It would be desirable to use weighting schemes for matches based on phylogenetic distances of genomes and annotation quality of genes. We used six different weighting criteria, three at the genome level and three at the gene level, and evaluated weighting criteria on all nine combinations of different weighting settings.

The first genome weight, called *equal-genome-weight*, assigned an equal unit weight to each reference genome. The second genome weight, called *phylo-genome-weight*, used phylogenetic distance of genomes using 16S rRNAs of reference genomes as a genome weight. By computing multiple sequence alignments of 16S rRNAs from target and reference genomes using CLUSTALW

(Thompson *et al*., 1994), distance d_i between a target genome and each reference genome G_i was obtained. We used $-\log(d_i)$ as a genome weight for a genome G_i. The third genome weight, called *quality-genome-weight*, considered the annotation quality of a reference genome, where the quality information was determined by counting at how many genes in the genome were in 'reviewed' status in UniProtKB/Swiss-Prot. The weight was set as a ratio of the number of genes with annotations where their status is reviewed divided by the total number of proteins in the genome.

The first weight at the 'gene level', called *equal-gene-weight*, was an uniform gene weight of 1. The second weight at the gene level, called *citation-gene-weight*, considered the number citations of a matching gene used for its annotation. We used reference record count—'RN' line in UniProtKB/Swiss-Prot. We assigned RN + 1 as a gene weight. A pseudo count of 1 was added to avoid division by zero and the weights were normalized. The third weight at the gene level, called *evidence-gene-weight*, was to utilize protein evidence (PE) code of each gene in UniProt. There are five types of evidence for the existence of a protein and PE was 1 for evidence at protein level, 2 for evidence at transcript level, 3 for inferred from homology, 4 for predicted and 5 for uncertain. (Protein existence, 2008). We used $1/2^{\mathrm{PE}}$ as a gene weight.

To evaluate the effect of the weighting schemes, we performed experiments using *Salmonella paratyphi A* as a target genome and 15 *Gammaproteobacteria* genomes as reference genomes. Out of the 15 reference genomes, 14 were chosen randomly. *Escherichia coli K12* was used as a reference genome to see an effect of using a well-annotated model organism as a reference genome. Figure 5 shows the results from the experiments.

The leftmost three boxes show a result when the equal-genome-weight scheme was used. The three boxes in the middle show a result when the phylo-genome-weight scheme was used. The rightmost three boxes show a result when the quality-genome-weight scheme was used. Left boxes (1st, 4th and 7th boxes) in each genome weighting scheme show a result of equal-gene-weight, middle boxes (2th, 5th and 8th boxes) in each genome weighting scheme show a result of citation-gene-weight and right boxes (3rd, 6th and 9th boxes) in each genome weighting scheme show a result of evidence-gene-weight.

We first discuss the effect of using genome weight. Increase in ACS using the phylo-genome-weight scheme ('16S' in the figure) compared with equal-genome-weight ('unit' in the figure) was because target genes had more homologs in close genomes and they were weighted higher than remote genomes. Using the quality-genome-weight scheme ('status' in the figure), ACS was further increased because genes in the well-annotated model organism had higher weights. In the experiment, the genome weight of *E.coli* was 1 since all genes in *E.coli* were reviewed and the next highest genome weight except *E.coli* was 0.2679. This resulted in increasing ACS of a gene annotation significantly when there were only a few matches including a match in *E.coli* because genes in *E.coli* had much higher genome weights.

Use of gene weight also increased ACS significantly. Use of the citation-gene-weight scheme ('bib' in the figure) increased ACS significantly than no-gene-weight ('unit' in the figure) in all three different genome weight settings. This was because the well-annotated model organism, *E.coli* in this experiment, had many citations for each gene. Use of the evidence-gene-weight scheme ('code' in the figure) further increased ACS in all three different genome weight settings. This showed that PE code was an effective, informative gene-level weight scheme for ACS.

4 CONCLUSIONS

New sequencing technologies have enabled small research labs to sequence genomes easily. However, annotation of genomes remains a significant challenge. The annotation scoring scheme, ACS, which we proposed in this article is a tool that can reduce genome annotation cost significantly. For example, users can use annotation services such as CMR (http://cmr.jcvi.org/tigr-scripts/CMR/CmrHomePage.cgi) at JCVI or IMG (Markowitz *et al.*, 2008) at JGI. Then the use of ACS can reduce the number of gene annotations that should be reviewed manually. In addition, reannotation of existing genomes can be speed-up reliably with use of ACS.

ACS effectively combines both sequence and textual similarity to denote the quality of annotation. Extensive experiments with many different reference genome sets demonstrated that ACS was effective to denote the annotation quality for reference genomes of various phylogeny and for varying number of genomes. ACS also can handle many issues in textual annotations such as use of abbreviations and synonyms. We found that quality weightings based on sequence homology, taxonomic distance and textual similarities all improved the overall score when used appropriately for the dataset as shown in Section 3.6.

A confidence measure like ACS depends on accurate and complete databases for quality information, such as MeSH and BioThesaurus, which are not as complete as desired. We have shown that combining several measures of quality information will improve the result. Missing information in one quality source can be made-up for by addition of other quality sources.

In general, this approach to summarize annotations from reference genomes as a quality score, derived from the three domains of sequence homology, taxonomic distance and textual similarities, can be extended to all kingdoms of life although our examples focused on microbial genomes.

As a future study, we are looking into improving ACS by using more resources that can be useful to generate confidence score, such as protein domain, protein structure, GO and other annotation quality measures on genes, e.g. abnormal protein analysis (Nagy *et al.*, 2008).

Funding: MetaCyt Microbial Systems Biology grant from the Lilly Foundation (in part); National Science Foundation, USA (grant MCB 0731950, in part).

Conflict of Interest: none declared.

REFERENCES

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

Berry,M.W. (2003) *Survey of Text Mining I: Clustering, Classification, and Retrieval*. Springer, New York, p. 26.

Centre for Horticulture and Lanscape, The Richards Function, The University of Reading, UK. Available at http://www.horticultureandlandscape.rdg.ac.uk/hlm_richards.htm (last accessed date November 11, 2009)

Chakrabati,S. (2002) *Mining the Web: Discovering Knowledge form Hypertext Data*, 1st edn. Morgan Kaufmann, San Fransisco, CA, p. 49.

Christiane,F. (1998) *WordNet: An Electronic Lexical Database*, The MIT Press, Cambridge, Massachusetts.

Huang,H. *et al*. (2003) iProClass: an integrated database of protein family, function and structure information. *Nucleic Acids Res.*, **31**, 390–392.

Iliopoulos,I. *et al*. (2003) Evaluation of annotation strategies using an entire genome sequence, *Bioinformatics*, **19**, 717–726.

Illunima Inc. (2007) DNA sequencing with Solexa technology. Available at http://www.illumina.com/downloads/SS_DNAsequencing.pdf (last accessed date December 10, 2008).

Liu,H. *et al*. (2006a) BioThesaurus: a web-based thesaurus of protein and gene names. *Bioinformatics*, **22**, 103–105.

Liu,H. *et al.* (2006b) Quantitative Assessment of Dictionary-based Protein Named Entity Tagging. *J. Am. Med. Inform. Assoc.*, **13**, 497–507.

Markowitz,V.M. *et al*. (2008) The integrated microbial genomes (IMG) system in 2007: data content and analysis tool extensions. *Nucleic Acids Res.*, **36** (Database issue), 528–533.

Mikkelsen,T.S. *et al.* (2005) Improving genome annotations using phylogenetic profile anomaly detection, *Bioinformatics*, **21**, 464–470.

Nagy,A. *et al.* (2008) Identification and correction of abnormal, incomplete and mispredicted proteins in public databases. *BMC Bioinform.*, **9**, 353.

NIH (2007) An Overview of MeSH. Available at http://www.nlm.nih.gov/mesh/presentations/EAHIL_krakow_2007_sep/mesh_overview/index.htm (last accessed date November 11, 2009).

Overbeek,R. *et al.* (1999) The use of gene clusters to infer functional coupling. *Proc. Natl Acad. Sci. USA*, **96**, 2896–2901.

Porter,M.F. (1980) An algorithm for suffix stripping. *Program*, **4**, 130–137.

Protein existence (2008) http://www.uniprot.org/manual/protein_existence (last accessed date December 10, 2008).

Richards,F.J. (1959) A flexible growth function for empirical use. *J. Exp. Bot.*, **10**, 290–300.

Sanger,F. and Coulson,A.R. (1975) A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J. Mol. Biol.*, **94**, 441–448.

Sanger,F. *et al.* (1977) DNA sequencing with chain-terminating inhibitors. *Proc. Natl Acad. Sci. USA*,**74**, 5463–5467.

Tatusov,R.L. *et al.* (1997) A genomic perspective on protein families. *Science*. **278**, 631–637.

Thompson,J.D. *et al.* (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **11**, 4673–4680.

Wicker,T.*et al.* (2006) 454 sequencing put to the test using the complex genome of barley. *BMC Genomics*, **7**, 275.