Bioinformatics, 36(7), 2020, 2251–2252
doi: 10.1093/bioinformatics/btz859
Advance Access Publication Date: 19 November 2019
Applications Note

Genome analysis

KofamKOALA: KEGG Ortholog assignment based on profile HMM and adaptive score threshold

Takuya Aramaki[1], Romain Blanc-Mathieu[1], Hisashi Endo[1], Koichi Ohkubo[1,2], Minoru Kanehisa[1], Susumu Goto[3] and Hiroyuki Ogata[1,*]

[1]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011, [2]Hewlett-Packard Japan Ltd., Koto-ku, Tokyo 136-8711 and [3]Database Center for Life Science, Research Organization of Information and Systems, Kashiwa, Chiba 277-0871, Japan

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on April 9, 2019; revised on October 2, 2019; editorial decision on November 14, 2019; accepted on November 16, 2019

Abstract

Summary: KofamKOALA is a web server to assign KEGG Orthologs (KOs) to protein sequences by homology search against a database of profile hidden Markov models (KOfam) with pre-computed adaptive score thresholds. KofamKOALA is faster than existing KO assignment tools with its accuracy being comparable to the best performing tools. Function annotation by KofamKOALA helps linking genes to KEGG resources such as the KEGG pathway maps and facilitates molecular network reconstruction.

Availability and implementation: KofamKOALA, KofamScan and KOfam are freely available from GenomeNet (https://www.genome.jp/tools/kofamkoala/).

Contact: ogata@kuicr.kyoto-u.ac.jp

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

1 Introduction

Automatic gene function annotation is an important first step to interpret genomic data. Kyoto Encyclopedia of Genes and Genomes (KEGG) is a widely used reference knowledge base, which helps investigate genomic functions by linking genes to biological knowledge such as metabolic pathways and molecular networks (Kanehisa and Goto, 2000). In KEGG, the KEGG Orthology (KO) database—a manually curated large collection of protein families (i.e. KO families)—serves as a baseline reference to link genes with other KEGG resources such as metabolic maps through *K* number identifiers. Currently, KOs are assigned to 12 934 525 (48%) protein sequences in the KEGG GENES database (27 173 868 proteins). Three existing tools, BlastKOALA, GhostKOALA (Kanehisa *et al.*, 2016) and KAAS (Moriya *et al.*, 2007), are available to assign KOs to protein sequences. These tools use homology search software such as BLAST (Altschul *et al.*, 1997) and GHOSTX (Suzuki *et al.*, 2014) to search amino acid sequences against GENES. To reduce the lengthy computational times required for multiple pairwise sequence comparisons, these tools use selected representative sequences from GENES to build their target database. In this study, we propose to employ profile hidden Markov model (pHMM) to compress the database and to define adaptive thresholds for similarity scores, which can be used for reliable KO assignments.

2 Implementation

For each set of protein sequences in GENES annotated with a given KO, we generate a pHMM as follows. First, sequence redundancy in the sequence set is reduced by CD-HIT (Fu *et al.*, 2012) with 100% sequence identity clustering cutoff. Next, MAFFT (Katoh *et al.*, 2005) and HMMER/hmmbuild (Eddy, 2008) are used to align sequences and to generate a pHMM, respectively.

A family-specific adaptive score threshold is computed for each KO family as follows. For a given KO family, non-redundant sequences belonging to the family are randomly divided into three groups. One of the groups is used as the positive training dataset, while the sequences in the remaining two groups are used to generate a pHMM. Sequences belonging to the remaining KO families serve as the negative training dataset for the KO under consideration. Sequence similarity scores (bit scores) between sequences in the positive/negative training datasets and the pHMM are computed using HMMER/hmmsearch. Based on the distributions of two sets of bit scores for the sequences in the positive and negative datasets, we determine a threshold score, T, which maximizes the F-measure (Supplementary Data). This procedure is repeated three times by replacing the positive training dataset among the three groups. Finally, the adaptive score threshold for the KO family is defined as the average of T ($\bar{T}$). $\bar{T}$ is used as a criterion to assign the KO family to new sequences.

Table 1. Comparison of the performance of KofamScan with other tools

	KofamScan	BlastKOALA	GhostKOALA	KAAS
Entire database (40 genomes)				
Precision	0.844	0.835	0.787	0.895
Recall	0.888	0.950	0.952	0.739
F	0.866	0.889	0.862	0.810
Prokaryote database (20 genomes)				
Precision	0.906	0.906	0.907	0.881
Recall	0.846	0.793	0.867	0.709
F	0.875	0.846	0.886	0.786

The database of HMMs with the adaptive score thresholds was named KOfam. When the present study was performed using KEGG release 88.2, KOfam contained 20 654 pHMMs. We developed KofamScan software and employed it in KofamKOALA webserver to annotate genes using KOfam and to link them with other KEGG resources through *K* numbers for versatile function investigation.

3 Assessment and discussions

To compare the performance of KofamScan with BlastKOALA, GhostKOALA and KAAS, we used 40 genomes (20 eukaryotes and 20 prokaryotes; Supplementary Table S1) randomly selected from 6030 genomes recorded in GENES as test queries. This test set contains 383 202 sequences (143 662 sequences with KOs) corresponding to 16 166 distinct KOs. From GENES, we removed all the genomes belonging to the genera of the selected 40 test query genomes. Then, using the remaining GENES sequences with KO annotations, we generated a test KOfam database for this assessment. As for BlastKOALA, GhostKOALA and KAAS, we used the default target databases used in their respective webservers after removing genomes from the genera that were represented by the test queries.

The KOfam database created for this assessment contained 20 394 pHMMs, of which 9414 were represented by prokaryotic sequences. For the 40 genomes constituting our test set, the performance (*F*-measure) was comparable among KofamScan (0.866), BlastKOALA (0.889) and GhostKOALA (0.862), while KAAS showed a lower *F*-measure (0.810) (Table 1). To perform another test using only 20 prokaryotic genomes as test queries, we reduced the target databases either by excluding pHMMs composed exclusively of eukaryotic sequences (for KofamScan) or by using the target database for prokaryotes (for BlastKOALA, GhostKOALA and KAAS). Again, the performance of KofamScan ($F = 0.875$) was comparable to BlastKOALA (0.846) and GhostKOALA (0.886), while KAAS showed the lowest *F*-measure (0.786).

Regarding CPU time, KofamScan was 69, 2.1 and 1.1 times faster than BlastKOALA, KAAS and GhostKOALA, respectively, when they were tested for the annotation of 40 genomes (Supplementary Tables S2 and S3). CPU time for this calculation was 2 h 26 m 18 s for KofamScan, while it was over 168 h for BlastKOALA. For the test with 20 prokaryote genomes, KofamScan was 83, 1.9 and 1.8 times faster than BlastKOALA, KAAS and GhostKOALA, respectively. Required CPU time was 11 m 59 s for KofamScan, while it was over 16 h for BlastKOALA. The latter result indicates that KofamScan can benefits more from the reduction of the target database compared to the three other tools while it is among the tools showing the highest *F*-measures.

We developed a database of pHMMs based on the KO and GENES databases. The adaptive score thresholds are pre-computed for individual KO families, and can be used to assign KOs (*K* numbers) to sequences using KofamScan and KofamKOALA. Sequence matches with scores above the thresholds are considered more reliable than other matches and thus highlighted with '*' marks in the output of these tools. KofamScan users are able to customize KOfam by choosing a subset of KOs so that they can focus on the annotation of specific class of proteins while reducing computational time. KofamKOALA webserver has additional functions to automatically send the search results to KEGG Mapper for reconstruction of pathways (PATHWAY), pathway modules (MODULE) and hierarchical function classifications (BRITE).

Acknowledgements

Computation time was provided by the SuperComputer System, Institute for Chemical Research, Kyoto University.

Funding

This work has been supported by the JSPS/MEXT/KAKENHI (Nos. 26430184, 18H02279, 16H06429, 16K21723, 16H06437) and the Collaborative Research Program of the Institute for Chemical Research, Kyoto University (No. 2018-30).

Conflict of Interest: none declared.

References

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.

Eddy,S.R. (2008) A probabilistic model of local sequence alignment that simplifies statistical significance estimation. *PLoS Comput. Biol.*, **4**, e1000069.

Fu,L. *et al.* (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**, 3150–3152.

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

Kanehisa,M. *et al.* (2016) BlastKOALA and GhostKOALA: KEGG tools for functional characterization of genome and metagenome sequences. *J. Mol. Biol.*, **428**, 726–731.

Katoh,K. *et al.* (2005) MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Res.*, **33**, 511–518.

Moriya,Y. *et al.* (2007) KAAS: an automatic genome annotation and pathway reconstruction server. *Nucleic Acids Res.*, **35**, W182–W185.

Suzuki,S. *et al.* (2014) GHOSTX: an improved sequence homology search algorithm using a query suffix array and a database suffix array. *PLoS One*, **9**, e103833.