GFClassify: Tool for Gene Family Detection and Classification

Jordan A. Fish   Tracy K. Teal   Yanni Sun
Tom Schmidt   James R Cole

March 3, 2013

1 Introduction

Classifying protein coding sequences into gene families is a difficult problem. A widely used program for protein homology search and classification is HMMER3[4]. However HMMER3 is designed for searching whole genomes for regions matching gene families; using HMMER3 to classify open reading frames requires careful tuning of inclusion thresholds in addition to building high quality training sets and alignments. Finally, HMMER3 doesn't allow for classifications to be mutually exclusive; ~~users must post process the output from HMMER3 to sort results into the highest scoring classes.~~

To facilitate easy gene family identification the tool GFClassify was written with the goal of classifying query sequences to exactly one of a set of input gene families. GFClassify takes nucleotide sequences as input so it can be used as a fast prefilter on raw reads. GFClassify utilizes interpolated context models[3] (ICMs) to classify query sequences in to one of a set of predefined classes. ICMs are used to predict the probability of a symbol (in this case a nucleotide) given the preceding k symbols and works well in separating differentially conserved sequences. For a detailed description of ICMs see [8] [3]. GFClassify can be used to distinguish between two or more paralogous gene familes (ex. Ammonia Monooxygonase [amoA] and Methane Monooxygenase [pmoA]) and distinguish between well defined groups in a protein family (ex. Nitrogenase Reductase [nifH]).

2 Methods

Two classification tasks, classifying sequences in to paralogous gene families and classifying a sequence in to one evolutionarily divergent class of a single gene family, were chosen to evaluate GFClassify. These two problems are biologically interesting expressions of a single probablistic task, determining which class assignment is most probable for an input sequence.

For classifying sequences in to paralogous gene familes amoA, pmoA or non-target classes were selected for testing. GFClassify was compared to HMMER3 for the task of distinguishing between paralgous gene families. To compared GFClassify to two common methods of gene family classification, Hidden Markov Models and nearest neighbor. To test the ability of GFClassify to label a gene sequence as one of several evoluationaily divergent classes nifH was chosen. GFClassify was compared to nearest neighbor classification in that test.

2.1 Gene Family Classification

GFClassify classifies query reads in to one of a set of input Interpolated Context Models (ICMs) by computing the probability that the ith ICM (ω_i) produced the query sequence.

$$P_i = P(\omega_i|x) = ICM_i(x)$$

GFClassify can, with the command line option -c, search the reverse complement of the query sequences in addition to the forward orientation. To choose the most probable orientation the probably of each input IMM producing the forward and reverse complement query is computed then the orientation is determined as the orientation that produces the highest probability of being produced from any of the input ICMs. Once the orientation has been selected, either by testing the forward and reverse query sequence or by assuming the query is in the correct orientation, the class ω_i with the highest probability, $\arg!\max_i Pi$, is selected as the potential label for the sequence. Then the log-likelihood ratio, ℓ, is computed between the highest, P_i, and second highest. P_j, probability ICMs as

$$\ell = P_i - P_j$$

If ℓ is greater than a user supplied threshold θ, the query is classified as class ω_i, otherwise the query is rejected as too close to decide. Included with

GFClassify are models for amoA, pmoA, and the four nifH families. Additionally a background model is included for determining whether a query sequence is from a non-target gene family. The construction of this background model is described in the ICM Building section below.

For each query sequence GFClassify reports the predicted orientation, log probability for each ICM and the label corresponding to the class prediction. If the log-likelihood ratio is less than the user specified threshold θ, the class label is replaced with "rejected". All probabilities are reported in log base e (nats) and if a classification threshold is supplied it too is expected to be in nats.

2.2 ICM Building

GFClassify uses Glimmer 3's ICM implementation and Glimmer's build-icm command was used to build the ICMs [2]. To find the optimal values for the tuning parameters for the ICM (window width, model depth and periodicy) each combination of a range of parameter values were used to build ICMs and the error rate was computed using 10-fold cross validation. The combination of parameters that yielded the lowest error rate were used to build ICMs for testing. The grid search utility to perform the parameter value search was written in python and is included with GFClassify.

Sequences used to build the amoA and pmoA models were obtained from the RDP's Functional Gene Pipeline (Fungene, http://fungene.cme.msu.edu). Additionally one of the testing sets for amoA and pmoA sequences were selected from Fungene. All sequences with genbank annotation of amoA or pmoA were downloaded to use as training sequences. The training sequences were then clustered with cdhit-est v4.5.4 with all default parameters at various percent similarities [6] to remove closely related sequences that would bias the models.

To be able to distinguish target from non-target sequences a background model was built from 10,000 randomly select protein sequences from all gene families in FunGene. This model represented the probabilities of seeing the words in a query sequence by random chance. In order to classify a sequence as one of the target classes the resulting score for any target model must be higher than the score from the background model.

Four ICMs for the five groups of nifH were built using reference sequences from Zehr's nifH Database [9]. NifH groups four and five are "nifH-like" and were grouped together in Zehr's database so the two groups were considered as one class in our analysis. There were 183 sequences from group one, 88 sequences from group two, 48 sequences from group three and 356 sequences from groups four and five in the nifH training set.

2.3 Testing GFClassify

GFClassify

2.3.1 *amoA/pmoA* Classification

HMMER3 is a widely used homology search tool that was used to assess the performance of GFClassify on differentiating amoA and pmoA. Both tools were tested on their ability to differentiate amoA sequences from pmoA sequences. To compare HMMER3 and GFClassify training error the same training sets were used and error rate was estimated with 10-fold cross validation. The HMMs used for comparing HMMER3 to GFClassify were built from 1,311 sequences from amoA and 479 sequences from pmoA. These sequences were selected at random from the sequences used to build the ICMs. The random downsampling (downsampled to 10%) to for generating the seed alignment for building the HMM. To make the seed alignments for HMMER3 the training sequences were aligned with MUSCLE[5], a multiple sequence alignment tool. Then a nucleotide HMM was built for amoA and pmoA using hmmbuild and ICMs for the two genes were built using build-icm.

GFClassify and HMMER3 were compared on their performance on a mock dataset selected from Fungene and five real amplicon datasets. The mock dataset contained 1,000 amoA and 1,000 pmoA sequences selected at random from Fungene and a set of 10,000 sequences randomly sampled across all gene families in Fungene. These sequences were then filtered to remove any that had been used to train the ICMs.

The amplicon datasets were unlabeled and contained both target and non-target reads. In addition to running GFClassify and HMMER3 these sequences were also annotated as amoA or pmoA based on a phylogenetic analysis. First the reads were compared to a reference set of amoA/pmoA sequences using BLAST[1] to filter out non-target reads. The sequences that had significant hits in the BLAST search were then aligned with reference amoA and pmoA sequences using MUSCLE. The aligned sequences were then used to build a tree in ARB[7]. Groups of sequences were labeled as amoA or pmoA based on the gene family of the closest reference sequence.

2.3.2 nifH Group Classification

The nifH reads for classification came from a set of NEON samples taken from soil in several states. The reads were filtered to remove all non nifH sequences before classification; the final sequence set contained 278,329 sequences.

The input reads were classified in to one of four groups (1, 2, 3 and 4-5) using GFClassify and a nearest neighbor approach. Each query was translated to protein and pairwise aligned to each of the sequences in the Zehr reference set. The pairwise alignment that produced the highest score was considered the nearest neighbor and classified in to the same nifH group as the reference sequence.

2.4 Implementation

GFClassify is written in C++ and the source code is available under the BSD licence on GitHub.

Two tools for model evaluation are included: a leave one out testing script (leave_one_out.py) and a k-fold cross validation script (cross_validate.py). Each takes two or more sets of training sequences (one per class) and performs the associated testing, reporting the error.

The grid search tool (grid.py) helps automate searching for optimal sets of parameters. The script takes a range of values for any combination of the three ICM tuning variables (context length, model depth, model period) and builds models with all combinations of values to find the one with the lowest error rate (using cross validation or evaluation on the training data).

A tool to help researchers identify good values for θ is included, find_threshold.py. Given the output of a labeled test set it, computes and reports the min, max, average and standard deviation of the smallest log likelihood score for all correctly classified and misclassified reads.

Cluster Distance	amoA Clusters	pmoA Clusters	GFClassify Error	HMMER3 Error
0.10	353	597	2.75%	3.01%
0.05	1169	1319	1.24%	1.60%
0.03	2125	1908	1.04%	1.32%
0.02	3138	2383	0.94%	1.32%
0.01	5426	3295	0.69%	-
0.00	13135	5448	0.46%	-

Figure 1: Training set selection based on cdhit clustering, error calculated as the average of 10-fold cross-validation errors

3 Results

3.1 Model Construction

3.1.1 amoA/pmoA

The seed sequences for building HMMs and ICMs were clustered to 6 distance cutoffs. Each set of sequences for each cutoff was used as a training dataset and the generalization error rate was estimated using 10-fold cross validation (Table 1).

GFClassify's performance was compared to HMMER3 by looking at both accuracy and time to train and test. The smallest training sets, cdhit clustered to 10% distance, were used in this comparison (Table 2). The build-icm command was used to build ICMs with a window size of 12, maximum depth of 7 and periodicy of 1; build-icm took approximately 90 seconds to train the amoA and pmoA models. HMMER3 took approximately 1 second to train both models; however, the multiple sequence alignment using MUSCLE to create the seed alignment for HMMER3 took approximately 600 seconds.

The nifH sequences used to build the ICMs for each group came from Zehr's nifH reference set; the generalization error rate estimate using 10-fold cross validation was approximately 20% due to the large difference in number of training samples per class.

Method	Dataset	amoA	pmoA	bg	Error Rate	Time (s)
GFClassify	Fungene Mock	984	982	10673	$6.0E^{-4}$	19.00
HMMER3	Fungene Mock	984	982	10673	$6.0E^{-4}$	423.63

Figure 2: GFClassify and HMMER3's performance on the Fungene mock dataset

Method	amoA	pmoA	bg	Error	Time (s)
GFClassify	104	977	977	0.4%	2.35[1]
HMMER3	105	992	961	0.4%	$(7.54 + 204)$[1]
Expected	104	984	970	-	-

Figure 3: GFClassify and HMMER3's performance on amplicon data using the hand curated BLAST+ARB method as the expected labels

[1] These datasets contained sequences in both forward and reverse orientation, GFClassify can search both strands HMMER3 cannot

3.2 amoA/pmoA Performance

To assess the performance of GFClassify and HMMER3 on different types of query sequences one labeled (Table 2) and four unlabeled (Table 3) test sets were run through GFClassify and HMMER3. The small amount of error in the mock dataset was expected since the background dataset contained a random sampling of protein sequences; some real amoA and pmoA sequences were present in the background dataset. The results of using BLAST to filter non-target sequences and using a phylogenetic approach to label sequences as amoA or pmoA are also reported. However, the time for this approach is not reported since it is only partially automated. The number of sequences classified into each category was recorded along with the running time.

3.3 Comparison to Nearest Neighbor

To benchmark the accuracy of GFClassify's labels on the NEON sequence set the labels were compared to the label of the nearest neighbor of the query in the reference set. The nifH group ICMs were built with a window size of 12, maximum depth of 7 and periodicy of 1. Out of 280,000 queries the labeling from GFClassify and Nearest Neighbor differred on 532 sequences and the ma-

Class	GFClassify	HMMER3	Nearest Neighbor
1	250308	249878	249871
2	27190	27582	27592
3	786	843	847
4-5	45	25	251
Time	2m	542m	690m
Error	0.2%	0.01%	-

Figure 4: Number of sequences with each nifH group label according to GFClassify and Nearest Neighbor using Nearest Neighbor classification as the expected label

GFClassify NifH Group	Expected Label			
	1	2	3	4-5
1	249,829	432	48	0
2	20	27,158	12	0
3	0	0	785	1
4-5	16	2	2	25

Figure 5: Confusion matrix for *nifH* group classification using GFClassify

jority of the differences was between classes 1 and 2 (Table 5). The number of sequences in each class is shown in Table 4. All sequences were within 20% distance to their nearest neighbor in the reference set.

4 Discussion

GFClassify allows researchers to easily and quickly filter amplicon sequences coming off sequencing machines based on whether they are a member of the target gene family. GFClassify classifies sequences faster than HMMER3 and the combined BLAST+ARB approach with comparable numbers of sequences assigned to each class. GFClassify is also built with mutually exclusive classifications in mind so no post-processing of results is required as with HMMER3.

The two classification methods evaulated as well as GFClassify all require care when developing training sets to ensure all the diversity is captured in the training set. GFClassify requires a larger training set than HMMER3 but is more resillient to noise in the training set.

HMMER3, GFClassify and the Nearest Neighbor classification approaches produced comparable results. HMMER3 tended to be more sensitive, however this sensitivity came at significantly increased classification time.

References

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, October 1990.

[2] Arthur L. Delcher, Kirsten A. Bratke, Edwin C. Powers, and Steven L. Salzberg. Identifying bacterial genes and endosymbiont dna with glimmer. *Bioinformatics*, 23(6):673–679, 2007.

[3] Arthur L. Delcher, Douglas Harmon, Simon Kasif, Owen White, and Steven L. Salzberg. Improved microbial gene identification with glimmer. *Nucleic Acids Research*, 27(23):4636–4641, 1999.

[4] Sean R. Eddy. Accelerated profile hmm searches. *PLoS Comput Biol*, 7(10):e1002195, 10 2011.

[5] Robert C. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004.

[6] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[7] Wolfgang Ludwig, Oliver Strunk, Ralf Westram, Lothar Richter, Harald Meier, Yadhukumar, Arno Buchner, Tina Lai, Susanne Steppi, Gangolf Jobb, Wolfram Frster, Igor Brettske, Stefan Gerber, Anton W. Ginhart, Oliver Gross, Silke Grumann, Stefan Hermann, Ralf Jost, Andreas Knig, Thomas Liss, Ralph Lmann, Michael May, Bjrn Nonhoff, Boris Reichel, Robert Strehlow, Alexandros Stamatakis, Norbert Stuckmann, Alexander Vilbig, Michael Lenke, Thomas Ludwig, Arndt Bode, and Karl-Heinz Schleifer. Arb: a software environment for sequence data. *Nucleic Acids Research*, 32(4):1363–1371, 2004.

[8] Steven L. Salzberg, Arthur L. Delcher, Simon Kasif, and Owen White. Microbial gene identification using interpolated markov models. *Nucleic Acids Research*, 26(2):544–548, 1998.

[9] Jonathan P. Zehr, Bethany D. Jenkins, Steven M. Short, and Grieg F. Steward. Nitrogenase gene diversity and microbial community structure: a cross-system comparison. *Environmental Microbiology*, 5(7):539–554, 2003.