World Academy of Science, Engineering and Technology
International Journal of Bioengineering and Life Sciences
Vol:4, No:10, 2010

MIM: A Species Independent Approach for Classifying Coding and Non-Coding DNA Sequences in Bacterial and Archaeal Genomes

Achraf El Allali, John R. Rose

Abstract—A number of competing methodologies have been developed to identify genes and classify DNA sequences into coding and non-coding sequences. This classification process is fundamental in gene finding and gene annotation tools and is one of the most challenging tasks in bioinformatics and computational biology. An information theory measure based on mutual information has shown good accuracy in classifying DNA sequences into coding and non-coding. In this paper we describe a species independent iterative approach that distinguishes coding from non-coding sequences using the mutual information measure (MIM). A set of sixty prokaryotes is used to extract universal training data. To facilitate comparisons with the published results of other researchers, a test set of 51 bacterial and archaeal genomes was used to evaluate MIM. These results demonstrate that MIM produces superior results while remaining species independent.

Keywords—Coding Non-coding Classification, Entropy, Gene Recognition, Mutual Information.

I. Introduction

The classification of DNA sequences as coding and non-coding is an ongoing research problem in bioinformatics. The production of new DNA sequences from genome projects has increased the need to analyze these new sequences and find new genes. The classification process into coding and non-coding sequences is part of the well studied microbial gene prediction problem. Current state of the art gene finders include GenemarkHMM programs [1], Prodigal [2], and Glimmer [3]. These programs obtain good accuracy in calling genes from raw genome data which leads some people to believe that microbial gene finding is a solved problem. However there are several issues with the predictions of these programs. The first problem is performance across the GC content spectrum. Most of these methods perform better in low GC genomes than high GC genomes. This is due to the fact that high GC content results in fewer candidate stop codons. This in turn results in longer candidate open reading frame (ORFs). This leads us to the second problem which is translation initiation site (TIS) prediction. Long ORFs have more candidate start codons which means that the problem of selecting the correct TIS is harder. There are several methods that post-process the results from gene finders in order to correct their TIS predictions, these programs include TriTISA [4] and GSFinder [5]. The last problem that gene finders encounter is the large number of predictions. This problem is hard to asses as the only way to know the total number of genes in a genome, hence the real precision of a gene finder, is by the hand curation of the entire genomes. One way to address this problem is to design more accurate classifiers capable of distinguishing coding from non-coding sequences and thus decreasing the number of false positives given by gene finders. In this paper we describe a high accuracy classification method capable of distinguishing coding from non-coding sequences which can be incorporated into any gene predictor.

There are three major classes of gene-finding tools designed around different types of information that can be used to distinguish coding from non-coding sequences: The first class is that of signal-based methods. These methods look for signals with functional significance such as signals in the vicinity of coding regions, translation initiation and termination, promoter regions, splice junctions, etc [6]–[10]. The second class includes comparison-based methods. These methods compare query sequences with known sequences in public databases using local alignment [11]. The third class consists of content-based methods. These methods use statistical features of both coding and non-coding sequences such as GpC islands, GC content, codon bias, nucleotide distribution, etc. [12]–[15]. Current gene finders use a combination to these methods to achieve better accuracy.

Content-based methods compute statistics that distinguish coding from non-coding DNA. These statistics are used as a measure of the likelihood of a sequence being a coding sequence. They can also be used as training data in pattern recognition systems that classify sequences as coding or non-coding. One method that is based on information found at the amino acid level is the mutual information method (MIM). Previous research has shown that a classifier based on MIM accurately classifies bacterial sequences while producing few false positives and false negatives [16]. In this paper we present an iterative approach based on our earlier MIM algorithm. Most methods require training data made up of pre-classified sequences from the genome that is being analyzed [17]–[19]. In contrast, the method described herein does not require any genome specific information but rather starts with an initial pre-computed characterization of coding and non-coding sequences. This universal characterization is used as a starting point for analyzing prokaryotic genomes. In particular, the method does not require pre-classified sequences from a genome being analyzed in order to evaluate the sequences from that genome. During the process of evaluating the sequences in a genome, MIM iteratively refines its characterization of

A. El Allali and J. R. Rose are with the Department of Computer Science and Engineering, University of South Carolina, Columbia South Carolina 29208. email: rose@email.sc.edu, elallali@email.sc.edu

coding and non-coding sequences. The experimental results show high classification accuracy for a set of 51 genomes used by other comparable methods [18]–[20]. Our classifier can be incorporated or used as a post processor to any genefinder in order to distinguish real coding sequences from false positives predicted by genefinders.

II. MATERIALS AND METHODS

A. Materials

We used a dataset of 60 complete bacterial genomes obtained from GenBank [21] in order to extract universal representation of the coding distribution. This dataset is based on Bern and Goldberg's list of 58 bacteria (2005) [22] which covered all bacterial phyla in GenBank as of December 2004. Starting with Bern and Goldberg's list, we have substituted *D. ethenogenes* for *Chloroflexus aurantiacus* and *R. baltica for Pirellula*. Additionally, we have added *Acidobacteria bacterium* and Magnetococcus from more recent publications, resulting in a database of 60 complete genomes. The list of these genomes and their accession numbers can be found in table II in the Appendix.

A second dataset of 51 complete bacterial and archaeal genomes is used for comparison with other methods. This dataset is used by Zhou's Fisher discriminant based method (FD) [19] and Guo-Sheng's global descriptor based method (GD) [20]. In order to support a direct comparison of MIM results with that of FD and GD, only coding and non-coding sequences with length greater than 300 bp were considered. A complete list of the categories, species names and the abbreviation of names, as well as the number of coding and non-coding sequences in these complete genomes was published by Zhou [18].

B. Universal coding and non-coding profiles for bacterial genomes

In our previous work, we derived the mutual information measure by first computing an amino-acid transition profile (AATP) for a genome based on known genes from that genome. The AATP profile measures the averaged frequency of all possible transitions of amino acid sequences provided by the known genes. There are 20 amino-acids plus "STOP", which means that the AATP profile contains 441 frequencies. Two AATP profiles are computed for sequences extracted for each genome. One is based on coding sequences and the other on non-coding sequences. The expectation is that the distribution of transitions in actual coding sequences is different from the distribution of transitions in non-coding sequences interpreted as coding. This step is referred to as the initial training phase. In order to generate the AATP profiles for coding sequences, we assembled representative sequences from the first dataset described in the previous section in order to form a universal coding set to be used in the initial training phase. The goal was to produce a seed training set that could be used to process a new genome without having to pre-classify any of the sequences in that new genome. The underlying hypothesis is that genes that have homologues in many of the 60 complete bacterial genomes used in this study are representative of prokaryotic genes. They should provide sufficient breadth to create a coding AATP profile that could serve as a starting point for the iterative MIM algorithm. We used BLASTCLUST [6] for the homology search. BLASTCLUST uses a single linkage measure to build clusters of sequences that satisfy the similarity criteria specified by the user. The following criteria were used in the clustering process: an E-value of 10^{-6} was used to establish sequence homology, a minimum of 60% sequence identity across at least 80% of the two sequences were used to identify matches. Since BLASTCLUST uses single linkage, each sequence will be present in only one cluster. The training dataset for coding sequences was compiled by arbitrarily selecting one member from each large cluster based on the following criteria: First, no genes from the genome under analysis are included. Second, The selected gene must come from a genome with comparable GC content as the genome being analyzed. The genomes were partitioned into three sets: genomes with GC content 30% or less, genomes with GC content in the range 31% to 50% and genomes with 51% or greater GC content 100%.

The non-coding training dataset was generated from the non-coding sequences in the second dataset following a leave-one-out approach in order to ensure the training data is independent of the input genome. The non-coding distribution contains non-coding sequences from all genomes from the same GC content range, except for the genome being analyzed. We arbitrarily select non-coding sequences until we obtain the same number of non-coding transitions as coding transitions.

C. The MIM algorithm

The basic Mutual Information Measure (MIM) is computed by the following algorithm:

1) Given the coding and non-coding training sequences. An amino acid transition profile is calculated separately for each dataset (coding and non-coding) in the form of a transition matrix. The matrix for the coding dataset is labeled $Trans_c$. The corresponding matrix for the non-coding dataset is labeled $Trans_{nc}$. Each entry, $Trans_c(i,j)$ for amino-acids i and j, contains the normalized frequency of the tuple ij in all coding sequences. $Trans_c$ is a 21 by 21 matrix. Equation (1) shows how $Trans_c$ is computed. Each entry in the matrix S_c is computed using equation (2) where t_{ij} denotes the number of occurrences of the tuple ij for each sequence s in the coding sequences. $Trans_{nc}$ is calculated in the same way using the non-coding sequences. This part of the algorithm is referred to as the training phase. To avoid null entries due to transitions not occurring in the training data, we initialize all entries in $Trans_c$ and $Trans_{nc}$ with a small epsilon value.

$$Trans_c(i,j) = \frac{S_c(i,j)}{\sum_{u=1}^{21}\sum_{v=1}^{21} S_c(u,v)} \tag{1}$$

$$S_c(i,j) = t_{ij} \tag{2}$$

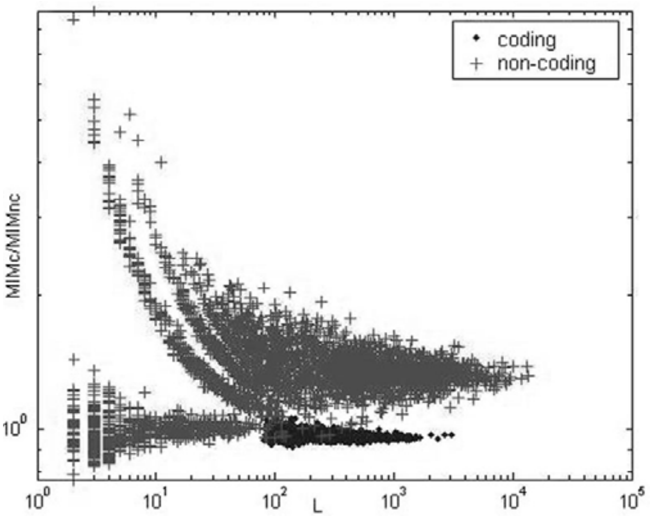

Fig. 1. Scatter plot of MIM_c/MIM_{nc} for all coding and non-coding sequences extracted from the *E. coli* genome.

2) MIM scores are calculated for a sequence "s" using both $Trans_c$ and $Trans_{nc}$ as follows:

$$MIM_c(i,j) = \sum_i \log(Trans_c(s(i), s(i+1))) \quad (3)$$

$$MIM_{nc}(i,j) = \sum_i \log(Trans_{nc}(s(i), s(i+1))) \quad (4)$$

In the above formulas, s(i) denotes the amino acid at the ith position of sequence "s"

3) A sequence "s" is classified as coding if $MIM_c/MIM_{nc} < 1$ otherwise it is classified as non-coding. Fig. 1. shows a scatter plot of MIM_c/MIM_{nc} as a function of the sequence length L for both coding and non-coding sequences extracted from the *Escherichia coli APEC O1* genome. We observe that the criterion $MIM_c/MIM_{nc} = 1$ serves as a good decision boundary for sequences of length greater than or equal to 80 amino acids. For most coding sequences in this length range the ratio MIM_c/MIM_{nc} is less than 1. Correspondingly, for most non-coding sequences in this length range the ratio MIM_c/MIM_{nc} is greater than or equal to 1.

III. Experimental Results

A. The iterative learning process

The iterative MIM algorithm pre-computes initial $Trans_c$ and $Trans_{nc}$ matrices using training data from a universal coding dataset and species-independent non-coding dataset as described in section II-B. The second phase is the iterative training phase. The MIM algorithm uses the initial $Trans_c$ and $Trans_{nc}$ matrices to classify all sequences in the input genome as coding or non-coding. These newly classified sequences are then used to compute a new $Trans_c$ matrix. In order to ensure the number of pseudo amino acid transitions in the non-coding dataset is the same as the number of amino acid transitions in the coding dataset, we keep the initial $Trans_{nc}$ matrix unchanged. This iterative process of sequence classification and computation of transition matrix continues until it converges or the maximum selected number of iterations is reached. The last phase is to classify all coding and non-coding sequences in the analyzed genome using the transition matrices resulted from the iterative procedure. In order to reduce the likelihood that an initial iteration includes missclassifications that could miss-train the classifier, we excluded those sequences close to the decision boundary by applying an exclusion band made of two user selectable decision boundaries, one for coding and one for non-coding. During the iterative portion of the algorithm, all sequences with scores within the exclusion band are excluded from computing the next $Trans_c$. Once the iteration process converges or times out, the exclusion band is discarded and the decision boundary $MIM_c/MIM_{nc} = 1$ is applied to classify all sequences.

After the iterative process is finished, we obtain the final coding transition matrix. For each sequence, two measures MIM_c and MIM_{nc} are computed using the resulted coding transition matrix and the original non-coding transition matrix respectively. If $MIM_c/MIM_{nc} < 1$ then the sequence is classified as coding, otherwise non-coding. Algorithm 1 gives the pseudo code for initial training, and the testing phases of the iterative MIM.

Algorithm 1 Pseudo code for IterativeMIM.

```
Given a genome $G$
Compute initial $Trans_c$ from universal coding
Compute $Trans_{nc}$ from non-coding sequences
repeat
  Classify all sequences in $G$ using the newly computed $Trans_c$ and the universal $Trans_{nc}$
  Compute new $Trans_c$ using the sequences classified as coding from previous step
until no changes in $Trans_c$ or maxiterations is reached
Classify all sequences in $G$ using the newly computed $Trans_c$ and the universal $Trans_{nc}$
Display the classification accuracy
```

B. Evaluation procedure

The majority of gene prediction methods use measures of prediction accuracy proposed by Burset & Guigo [23]. Once a sequence is classified, each nucleotide in this sequence will belong to one of the four categories: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). A true positive is a coding nucleotide that belongs to a correctly predicted sequence. A false positive is a non-coding nucleotide that belongs to an incorrectly predicted sequence. A true negative is a non-coding nucleotide that belongs to a correctly predicted sequence. A false negative is a coding nucleotide that belongs to an incorrectly predicted sequence. After testing all the sequences, we add up all the TP, FP, TN and FN measures from all sequences. These values are then used to derive measures of coding/non-coding sensitivity and specificity, as well as correlation.

1) Sensitivity:

Coding:

$$S_n = \frac{TP}{TP + FN} \tag{5}$$

Non-coding:

$$S_q = \frac{TN}{TN + FP} \tag{6}$$

2) Specificity:

Coding:

$$S_p = \frac{TP}{TP + FP} \tag{7}$$

Non-coding:

$$S_r = \frac{TN}{TN + FN} \tag{8}$$

3) Correlation Coefficient:

$$CC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FP) \times (TN + FN)}} \tag{9}$$

4) Approximation Correlation:

$$AC = \frac{1}{2}(S_n + S_p + S_q + Sr) - 1 \tag{10}$$

Two recently published sequence classification methods have adopted different measures of coding and noncoding sequence accuracies [14] [15]:

1) Coding Score:

$$a_c = \frac{\textit{No. of all correct coding discriminations}}{\textit{No. of coding sequences in the testing set}} \tag{11}$$

2) Non-coding Score:

$$a_{nc} = \frac{\textit{No. of all correct non-coding discriminations}}{\textit{No. of non-coding sequences in the testing set}} \tag{12}$$

We are including these measures to allow a direct comparison of MIM results with that of these approaches.

IV. Results

In order to evaluate the iterative MIM algorithm, we analyzed the 51 genomes of the test dataset described in the materials section. Table I reports the accuracy scores a_c and a_{nc} for MIM as well as the GD [14]and the FD [14] methods. MIM outperforms both GD and FD on the majority of the 51 organisms. The average and standard deviation of accuracy scores for all three methods at the bottom of Table I. Notice that the GD method has a high score of 96.77% accuracy for coding but a low 78.57% accuracy for non-coding, while the FD method has 92.77% score for non-coding but a low 80.86% score for coding. Iterative MIM on the other hand scores high averages on both coding and non-coding: 98.77% and 99.33% respectively.

Table III in the Appendix reports the Burset & Guigo classification scores. The average approximate correlation (AC) score for this set of genomes is 97.45%, with a standard deviation of 2.8%. For most genomes, the iterative MIM algorithm correctly distinguishes coding from non-coding sequences with AC score exceeding 99%. These results support the hypotheses that the iterative MIM algorithm is able to accurately classify sequences in a genome without requiring pre-classified sequences from that genome.

V. Discussion

In this paper, we present an iterative MIM algorithm that is capable of distinguishing coding from non-coding sequences in bacterial and archael genomes with high accuracy. Using a set of representative genes, we create initial coding transition matrices. The algorithm uses these initial transition matrices to classify all of the sequences in the genome that is being analyzed. Then a new coding transition matrix is calculated from the classification results while the non-coding transition matrix is held unchanged. The algorithm then alternates between sequence classification and transition matrix calculation. The algorithm iterates until converging or the specified maximum number of iterations is attained. Although we do not have a proof of convergence, empirically the iterative process required at most 3 iterations in the case of the 51 genomes in the test dataset. We allow the user to set a maximum number of iterations in order to guarantee termination. The results demonstrate that it is possible to accurately classify sequences in a bacterial or archaeal genome as coding or non-coding without requiring a subset of pre-classified sequences from that genome.

The accuracy of our algorithm depends on the correctness of the initial training data. The universal coding sequences are derived from an analysis of homologous sequences. Thus there is good reason to believe that these sequences are correctly identified as actual coding sequences. However, in order to evaluate the accuracy of our algorithm, we must rely on published annotations, many of which are generated automatically and have not been fully verified. For example, if our algorithm determines that a sequence is a non-coding sequence, but the annotation indicates that it is coding then this counts as a false negative. It could be that the annotation is incorrect and our algorithm is correct. However, since we are treating the annotations as ground truth, any mistakes in annotation that disagree with our algorithm's results will cause us to evaluate the results of our algorithm as producing more classification errors that may in fact be the case. At present our algorithm is achieving $\tilde{9}9\%$ accuracy for most genomes. Annotation errors make it unlikely that 100% accuracy can be achieved unless our algorithm makes the same errors as in the annotation. In order to illustrate this problem, we examined the *ecoli* genome since it has the maximum number of experimentally verified genes in bacteria. There are 881 verified genes hosted by the EcoGene database [24]. The iterative MIM algorithm was tested on the verified set after the algorithm has learned the coding transition matrix and without restricting the length limit on the input. The algorithm correctly classified 880 genes, the misclassified gene was the shortest gene with only 26 residues. This suggests that iterative MIM is very accurate in correctly identifying true coding sequences. In order to explore the validity of the false negatives produced by iterative MIM,

TABLE I
CODING AND NON-CODING SCORES FOR ITERATIVEMIM, GLOBAL DESCRIPTOR [19] AND FISHER'S DISCRIMINANT [20] METHODS

Methods	**MIM**		**GD**		**FD**	
Species	a_c	a_{nc}	a_c	a_{nc}	a_c	a_{nc}
Archaeoglobus fulgidus DSM 4304	**99.24**	**100**	96.64	64.00	85.68	92.33
Pyrococcus abyssi	**99.59**	**99.60**	98.52	78.57	86.16	94.81
Pyrococcus horikoshii OT3	**95.48**	**98.96**	90.15	60.32	79.55	78.63
M. jannaschii DSM4304	**99.48**	**100**	98.35	80.00	78.93	93.26
Halobacterium sp NCR-1	98.71	90.50	**99.15**	83.72	75.97	**95.50**
Thermoplasma acidophilum	**99.42**	**100**	95.67	75.00	81.42	93.48
Thermoplasma volcanium GSS1	**100**	**99.76**	96.75	72.50	80.72	90.56
M. thermoautotrophicum deltaH	**99.45**	**100**	99.09	68.75	85.83	96.36
Aeropyrum pernix	**99.47**	**100**	99.24	78.82	72.91	80.56
Sulfolobus solfataricus	**99.71**	**100**	94.35	84.85	77.01	87.44
Mycobacterium turberculosis H37Rv	98.06	**99.55**	**99.32**	81.58	81.25	98.16
Mycobacterium turberculosis CDC1551	96.20	**98.01**	**99.04**	64.44	83.29	92.03
Mycobacterium leprae TN	**97.81**	**100**	80.62	75.97	75.60	85.84
Mycoplasma pneumoniae M129	**98.84**	**99.11**	93.94	80.00	84.56	86.02
Mycoplasma genitalium G37	92.38	**100**	94.44	60.00	**95.74**	71.19
Mycoplasma pulmonis	96.40	**100**	**99.30**	94.74	82.80	91.45
Ureaplasma urealyticum (serovar 3)	**100**	**100**	96.43	88.89	95.12	89.22
Bacillus subtilis 168	**99.53**	**100**	99.03	81.40	81.12	89.33
Bacillus halodurans C-125	**99.86**	**100**	98.87	82.73	76.54	97.57
Lactococcus lactis IL 1403	**99.11**	**99.78**	97.77	82.00	75.33	95.64
Streptococcus pyogenes M1	**98.94**	**99.79**	98.01	77.55	81.47	93.11
Streptococcus pneumoniae	**98.32**	**99.81**	94.72	68.33	77.33	78.89
Staphylococcus aureus N315	**98.61**	**100**	97.82	88.24	76.11	83.48
Staphylococcus aureus Mu50	**98.78**	**100**	96.62	86.41	76.61	91.09
Clostridium acetobutylicum ATCC824	**99.63**	**99.65**	99.08	76.74	82.04	94.97
Aquifex aeolicus VF5	**99.60**	**95.59**	96.64	71.43	96.39	88.57
Thermotoga maritima MSB8	**99.59**	**99.30**	98.22	53.33	94.59	88.89
Chlamydia trachomatis (serovar D)	**99.64**	**100**	97.60	83.33	85.23	94.37
Chlamydia pneumoniae CWL029	**99.28**	**100**	97.42	72.00	75.36	97.65
Chlamydia pneumoniae AR39	**99.28**	**100**	96.88	80.95	79.78	95.17
Chlamydia pneumoniae J138	**98.98**	**100**	**98.98**	79.17	76.64	97.08
Synechocystis sp. PCC6803	**99.35**	**99.74**	98.80	85.90	77.60	96.86
Nostoc sp. PCC6803	**99.64**	**99.68**	97.09	84.68	69.18	98.6
Borrelia burgdorferi B31	**100**	**100**	97.42	**100**	94.15	92.86
Treponema pallidum Nichols	97.17	**100**	**99.4**6	50.00	82.13	99.33
Rhizobium sp. NGR234	98.30	**99.24**	**100**	60.71	67.86	89.55
Sinorhizobium meliloti	99.03	**99.54**	**99.19**	93.33	76.08	98.60
Caulobacter crescentus	97.66	97.81	**99.13**	94.00	79.68	**98.96**
Rickettsia prowazekii Madrid	**99.48**	**100**	90.15	96.32	82.04	**100**
Neisseria meningitidis MC58	**98.80**	**99.59**	96.77	77.63	69.46	92.10
Neisseria meningitidis Z2491	**99.53**	**99.08**	87.29	87.50	69.88	93.58
Escherichia coli K-12 MG1655	**99.33**	**100**	97.95	84.00	78.11	96.52
Escherichia coli O157:H7 EDL933	**99.13**	**98.25**	96.98	78.87	77.10	93.08
Haemophilus influenzae Rd	**99.28**	**99.41**	95.13	83.78	78.20	95.89
Xylella fastidiosa 9a5c	**96.57**	**99.65**	95.57	78.82	72.20	81.69
Pseudomonas aeruginosa PA01	99.07	96.62	**99.33**	89.09	82.68	**99.46**
Pasteurella multocida PM70	**99.16**	**100**	98.16	80.00	86.10	96.33
Buchnera sp APS	**99.81**	**100**	98.10	92.86	81.12	89.33
Agrobacterium tumefaciens	98.01	**99.04**	**99.01**	84.00	81.47	96.05
Helicobacter pylori 26695	**99.00**	**100**	97.85	73.08	85.35	91.29
Campylobacter jejuni	**99.62**	**98.81**	93.34	57.14	96.68	96.98
Average	**98.77**	**99.33**	96.77	78.57	80.86	92.15
S.D.	**0.26**	**0.84**	2.33	4.85	7.78	3.29

we analysed the *Rhodopirellula baltica* SH 1 genome. The first observation is that all the false negatives in this genome are for genes labeled as coding for hypothetical proteins. We analyzed these sequences using BLAST [11] to search for homologues in the entire microbial database hosted by NCBI. Only 3.27% of these sequences have BLAST hits outside of the *Rhodopirellula baltica* SH 1 genome and of these none have significant E-values. The fact that the vast majority of these sequences are annotated as genes coding for hypothetical proteins and do not have any known homologues suggests that they may in fact be non-coding sequences.

The annotations used to evaluate iterative MIM are from the Genbank [21] database at NCBI. Genbank relies on programs such as GenmarkHMM programs [1], Prodigal [2], Glimmer [3] and others for their gene annotations. The fact that our method successfully classifies verified genes and rejects many hypothetical genes that have low or no homology in the entire microbial database hosted by NCBI indicates that iterative MIM can serve as a better classification method for use in gene finding in order to improve their precision.

VI. Conclusion

Mutual information captures an important aspect of proteins, the interdependence between adjacent amino acids. Differences in mutual information measures for known coding and non-coding sequences can be used to classify unclassified sequences as coding or non-coding. The results presented in this paper support our hypothesis that the universal amino acid profiles we have derived can be used to provide sufficient breadth to create a coding and non-coding AATP profiles that serves as a starting point for an iterative MIM algorithm. We evaluated this algorithm on a set of 51 bacterial and archaeal genomes. This method demonstrates high accuracy in classify coding and non-coding sequences. Furthermore, it does not require pre-classified sequences from the target genome. Iterative MIM is thus a powerful species-independent method for classifying coding and non-coding DNA sequences which can be incorporated into any of the existing gene finders or used to post-process their classifications.

Acknowledgments

The experiments were run on an SGI Altix 4700 system, with 128 computing cores and 256GB shared memory funded by NSF award #0708391.

References

[1] A. Lukashin and M. Borodovsky, "Genemark.hmm: new solutions for gene finding." *Nucleic Acids Res.*, vol. 26, pp. 1107–1115, 1998.

[2] D. Hyatt, G.-L. Chen, P. F. LoCascio, M. L. Land, F. W. Larimer, and L. J. Hauser, "Prodigal: prokaryotic gene recognition and translation initiation site identification," *BMC Bioinformatics*, vol. 11, 2010.

[3] A. Delcher, K. Bratke, E. Powers, and S. Salzberg, "Identifying bacterial genes and endosymbiont dna with glimmer," *Bioinformatics*, vol. 23, pp. 673–679, 2007.

[4] G.-Q. Hu, X. Zheng, H.-Q. Zhu, and Z.-S. She, "Prediction of translation initiation site with tritisa," *Bioinformatics*, vol. 25, pp. 123–125, 2009.

[5] H. Ou, F. Guo, and C. Zhang, "Gs-finder: a program to find bacterial gene start sites with a self-training method," *Int. J. Biochem. Cell Biol.*, vol. 36, pp. 535–544, 2004.

[6] I. Rogozin and L. Milanesi, "Analysis of donor splice signals in different organisms," *J. Mol. Evl.*, vol. 45, pp. 50–59, 1997.

[7] J. Kleffe, K. Hermann, W. Vahrson, B. Wittig, and V. Brendel, "Logitlinear models for the prediction of splice sites in plant pre-mrna sequences," *Nucleic Acids Res.*, vol. 24, pp. 4709–4718, 1996.

[8] S. Brunak, J. Engelbrecht, and S. Knudsen, "Prediction of human mrna donor and acceptor sites from the dna sequence," *J. Mol. Biol.*, vol. 220, pp. 49–65, 1991.

[9] S. M. Hebsgaard, P. G. Korning, N. Tolstrup, J. Engelbrecht, P. Rouz, and S. Brunak, "Splice site prediction in arabidopsis thaliana pre mrna by combining local and global sequence information," *Nucleic Acids Res.*, vol. 24, pp. 3439–3452, 1996.

[10] M. Q. Zhang and T. G. Marr, "A weight array method for splicing signal analysis," *Comput. Appl. Biosci.*, vol. 9, pp. 499–509, 1993.

[11] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, pp. 403–410, 1990.

[12] P. McCaklon and P. Argos, "Oligopeptide biases in protein sequences and their use in predicting protein coding regions in nucleotide sequences," *Proteins: Structure, Function and Genetics*, vol. 4, pp. 99–122, 1988.

[13] R. Staden and A. D. McLachlan, "Codon preferences and its uses in identifying protein coding regions in long dna sequences," *Nucleic Acids Res.*, vol. 10, pp. 141–156, 1982.

[14] A. S. Kolaskar and B. V. B. Reddy, "A method to locate protein sequences in dna and prokaryotic systems," *Nucleic Acids Res.*, vol. 13, pp. 185–194, 1985.

[15] R. D. Blake and S. Early, "Distribution and evolution of sequence characterisitcs in e. coli genome," *J. Biomol. Struct. Dynam.*, vol. 4, pp. 291–307, 1996.

[16] J. R. Rose and A. El Allali, "Mutual information measure for distinguishing coding and non-coding dna sequences," *Biocomp*, vol. 1, pp. 214–219, 2008.

[17] Z. Ouyang and Z. S. She, "Multivariate entropy distance method for distinguishing coding and non-coding dna sequences," *J. Bioinform. Comput. Biol.*, vol. 2, pp. 353–373, 2004.

[18] L. Q. Zhou, Z. G. Yu, J. Q. Deng, V. Anh, and S. C. Long, "A fractal method to distinguish coding and non-coding sequences in a complete genome based on a number sequence representation, j," *Theor. Biol.*, vol. 232, pp. 559–567, 2004.

[19] Y. Zhou, L. Q. Zhou, Z. G. Yu, and V. V. Anh, "Distinguish coding and noncoding sequences in a complete genome using fourier transform," *International Conference on Natural Computation*, pp. 295–299, 2007.

[20] V. A. Guo-Sheng and Y. Zu-Guo, "Distinguishing coding from non-coding sequences in prokaryote complete genome based on the global desciptor," *IEEE Computer Society: Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 42–46, 2009.

[21] D. A. Benson, I. Karsch-Mizrachi, D. Lipman, J. Ostell, and E. Sayers, "Genbank," *Nucleic Acids Res.*, vol. 37(Database issue), pp. D26–31, 2009.

[22] M. W. Bern and D. Goldberg, "Automatic selection of representative proteins for bacterial phylogeny," *BMC Evolutionary Biology*, vol. 5, 2005.

[23] M. Burset and R. Guigo, "Evaluation of gene structure prediction programs," *Genomics*, vol. 34, pp. 353–367, 1996.

[24] R. K.E., "Ecogene: a genome sequence database for escherichia coli k-12," *Nucleic Acids Res.*, vol. 28, pp. 60–64, 2000.

Appendix

TABLE II. List of all 60 genomes and their accession numbers.

NC_003103	*Rickettsia conorii str. Malish* 7
NC_003098	*Streptococcus pneumoniae* R6
NC_002696	*Caulobacter crescentus* CB15
NC_000911	*Synechocystis sp.* PCC 6803
NC_002932	*Chlorobium tepidum TLS*
NC_004193	*Oceanobacillus iheyensis* HTE831
NC_004347	*Shewanella oneidensis* MR-1
NC_004463	*Bradyrhizobium japonicum* USDA 110
NC_004741	*Shigella flexneri 2a str.* 2457T
NC_004757	*Nitrosomonas europaea* ATCC 19718
NC_005071	*Prochlorococcus marinus str.* MIT 9313
NC_005070	*Synechococcus sp.* WH 8102
NC_004556	*Xylella fastidiosa Temecula1*
NC_004557	*Clostridium tetani* E88
NC_003272	*Nostoc sp.* PCC 7120
NC_00329	*Ralstonia solanacearum* GMI1000
NC_005027	*Rhodopirellula baltica* SH 1
NC_002929	*Bordetella pertussis* Tohama I
NC_005090	*Wolinella succinogenes* DSM 1740
NC_005125	*Gloeobacter violaceus* PCC 7421
NC_005296	*Rhodopseudomonas palustris* CGA009
NC_002939	*Geobacter sulfurreducens* PCA
NC_000964	*Bacillus subtilis subsp.* 168
NC_006360	*Mycoplasma hyopneumoniae* 232
NC_000962	*Mycobacterium tuberculosis* H37Rv
NC_002678	*Mesorhizobium* loti MAFF303099
NC_002936	*Dehalococcoides ethenogenes* 195
NC_003155	*Streptomyces avermitilis* MA-4680
NC_002179	*Chlamydophila pneumoniae* AR39
NC_004307	*Bifidobacterium longum* NCC2705
NC_002971	*Coxiella burnetii* RSA 493
NC_007519	*Desulfovibrio desulfuricans* G20
NC_002973	*Listeria monocytogenes str.* 4b F2365
NC_008009	*Acidobacteria bacterium* Ellin345
NC_002516	*Pseudomonas aeruginosa* PAO1
N_008576	*Magnetococcus sp.* MC-1
NC_001318	*Borrelia burgdorferi* B31
NC_000117	*Chlamydia trachomatis* D/UW-3/CX
NC_007146	*Haemophilus influenzae* 86-028NP

continued on next page

continued from previous page	
NC_000918	*Aquifex aeolicus* VF5
NC_000921	*Helicobacter pylori* J99
NC_000913	*Escherichia coli* K12
NC_003869	*Thermoanaerobacter tengcongensis* MB4
NC_000853	*Thermotoga maritima* MSB8
NC_002662	*Lactococcus lactis subsp. lactis* Il1403
NC_002950	*Porphyromonas gingivalis* W83
NC_003116	*Neisseria meningitidis* Z2491
NC_004663	*Bacteroides thetaiotaomicron* VPI-5482
NC_003317	*Brucella melitensis* 16M chromosome I
NC_002745	*Staphylococcus aureus subsp.* aureus N315
NC_006958	*Corynebacterium glutamicum* ATCC 13032
NC_001263	*Deinococcus radiodurans R1* chromosome1
NC_005085	*Chromobacterium violaceum* ATCC 12472
NC_003454	*Fusobacterium nucleatum subsp.* nucleatum ATCC 25586
NC_002505	*Vibrio cholerae* O1 biovar eltor str. N16961 chromosome I
NC_003902	*Xanthomonas campestris pv. campe stris str.* ATCC 33913
NC_002163	*Campylobacter jejuni subsp. jejuni* NCTC 11168
NC_000919	*Treponema pallidum subsp. pallidum str.Nichols*
NC_002162	*Ureaplasma parvum serovar 3 str.* ATCC700970
NC_005823	*Leptospira interrogans serovar Cope nhageni str. Fiocruz* L1-130 chromosome I

TABLE III
CLASSIFICATION SCORES FOR ALL 51 GENOMES.

Genomes	**Sn**	**Sp**	**Sq**	**Sr**	**CC**	**AC**
Archaeoglobus fulgidus DSM 4304	99.70	100	100	96.46	98.07	98.08
Pyrococcus abyssi	99.88	99.96	99.60	98.84	99.14	99.14
Pyrococcus horikoshii OT3	97.75	99.89	98.80	80.32	88.00	88.38
M. jannaschii DSM4304	99.80	100	100	98.40	99.10	99.10
Halobacterium sp NCR-1	99.17	99.04	92.53	93.52	92.12	92.12
Thermoplasma acidophilum	99.80	100	100	98.75	99.27	99.27
Thermoplasma volcanium GSS1	99.98	99.98	99.90	99.90	99.87	99.87
M. thermoautotrophicum deltaH	99.78	100	100	97.13	98.44	98.45
Aeropyrum pernix	99.80	100	100	98.20	99.00	99.00
Sulfolobus solfataricus	99.86	100	100	99.48	99.67	99.67
Mycobacterium turberculosis H37Rv	99.12	99.94	99.28	90.77	94.48	94.56
Mycobacterium turberculosis CDC1551	98.61	99.55	95.48	87.01	90.25	90.33
Mycobacterium leprae TN	99.10	100	100	99.55	99.33	99.33
Mycoplasma pneumoniae M129	99.27	99.43	99.51	99.37	98.79	98.79
Mycoplasma genitalium G37	97.41	100	100	97.41	97.41	97.41
Mycoplasma pulmonis	98.92	100	100	99.23	99.08	99.08
Ureaplasma urealyticum (serovar 3)	100	100	100	100	100	100
Bacillus subtilis 168	99.78	100	100	98.38	99.08	99.08
Bacillus halodurans C-125	99.96	99.99	99.93	99.78	99.83	99.83
Lactococcus lactis IL 1403	99.42	99.98	99.84	96.63	97.93	97.93
Streptococcus pyogenes M1	99.63	99.90	99.62	98.53	98.84	98.84
Streptococcus pneumoniae	99.00	99.98	99.92	95.76	97.31	97.33
Staphylococcus aureus N315	99.71	100	100	98.83	99.27	99.27
Staphylococcus aureus Mu50	99.54	100	100	98.10	98.82	98.82
Clostridium acetobutylicum ATCC824	99.89	99.93	99.63	99.38	99.41	99.41
Aquifex aeolicus VF5	99.78	99.70	96.60	97.50	96.80	96.80
Thermotoga maritima MSB8	99.86	99.96	99.44	97.97	98.61	98.62
Chlamydia trachomatis (serovar D)	99.40	100	100	94.50	96.92	96.95
Chlamydia pneumoniae CWL029	99.56	100	100	96.94	98.24	98.25
Chlamydia pneumoniae AR39	99.51	100	100	96.22	97.85	97.87
Chlamydia pneumoniae J138	99.78	100	100	97.96	98.87	98.87
Synechocystis sp. PCC6803	99.71	99.98	99.83	97.71	98.61	98.61
Nostoc sp. PCC6803	99.85	99.93	99.75	99.43	99.48	99.48
Borrelia burgdorferi B31	99.80	100	100	96.53	98.15	98.17
Treponema pallidum Nichols	98.02	99.97	99.56	77.20	86.78	87.38
Rhizobium sp. NGR234	99.20	99.91	99.26	93.74	96.03	96.06
Sinorhizobium meliloti	99.56	99.93	99.57	97.22	98.13	98.14
Caulobacter crescentus	99.15	99.79	97.29	90.00	93.05	93.11
Rickettsia prowazekii Madrid	99.94	100	100	99.88	99.91	99.91
Neisseria meningitidis MC58	99.45	99.72	99.24	98.54	98.48	98.48
Neisseria meningitidis Z2491	99.73	99.80	99.50	99.31	99.18	99.18
Escherichia coli K-12 MG1655	99.68	100	100	96.24	97.94	97.96
Escherichia coli O157:H7 EDL933	99.40	99.65	97.84	96.29	96.58	96.58
Haemophilus influenzae Rd	99.42	99.70	98.64	97.40	97.58	97.58
Xylella fastidiosa 9a5c	97.88	99.88	99.52	92.18	94.70	94.73
Pseudomonas aeruginosa PA01	99.45	99.65	96.64	94.85	95.29	95.30
Pasteurella multocida PM70	99.68	100	100	97.11	98.39	98.40
Buchnera sp APS	99.78	100	100	98.60	99.19	99.19
Agrobacterium tumefaciens	99.11	99.92	99.23	91.76	94.95	95.01
Helicobacter pylori 26695	99.59	100	100	96.40	97.98	97.99
Campylobacter jejuni	99.66	99.94	99.52	97.33	98.22	98.22
Average	99.427	99.90	99.32	96.25	97.42	97.45
S.D.	0.58	0.181	1.38	4.51	2.89	2.81