Identification of Prokaryotic Small Proteins using a Comparative Genomic Approach

Josue Samayoa, Fitnat Yildiz, Kevin Karplus

May 19, 2009

Abstract

Accurate prediction of genes encoding small proteins (on the order of 50 amino acids or less) remains an elusive open problem in bioinformatics. Some of the best methods for gene prediction use either sequence composition analysis or sequence similarity to a known protein coding sequence. These methods often fail for small proteins, however, either due to a lack of experimentally verified small protein coding genes or due to the limited statistical significance of statistics on short sequences.

Our approach is based upon the hypothesis that true short proteins will be under selective pressure for encoding the particular amino acid sequence, for ease of translation by the ribosome, and for structural stability. This stability can be achieved either independently or as part of a larger protein complex. Given this assumption, it follows that small proteins should display conserved local protein structure properties much like larger proteins. Our method incorporates neural-net predictions for 3 local structure alphabets within a comparative genomic approach to generate predictions for whether or not a given open reading frame encodes for a short protein. We have applied this method to the complete genome for *E. coli* strain K12 and looked at how well our method performed on a set of 60 experimentally verified small proteins from this organism. Out of a total of 11,467 possible ORFs, we found that 4 of the top 10 and 24 of the top 100 predictions belonged to the set of 60 experimentally verified short proteins. We also tested our method against all annotated ncRNAs in *E. coli* K12 and found that the best scoring ncRNAs by our method were significantly enriched for regulatory sRNAs that are complementary to protein-coding ORFs. Overall, our method represents a significant improvement over the state of the art for predicting true small protein encoding sequences.

1 Introduction

Increasingly, short proteins have been found to be important functional elements in cellular biology. Members of this class of molecules have been associated with a diverse set of functions including the regulation of amino acid metabolism [24], iron-homeostasis [21], spore development [18, 5] and antimicrobial activity [8]. Despite the recent discoveries of functionally relevant short proteins, there is still relatively little known about how widespread and critical short proteins are. Furthermore, the advent of next-generation sequencing technologies has enabled transcriptional profiling of complete genome sequences [4]. In the yeast genome, these efforts have led to the expansion of the transcriptome [14]. The problem of discriminating short protein-encoding sequences from other short RNAs will only increase as transcriptional profiling efforts expand.

Due to the lack of introns and alternative splicing mechanisms, prokaryotic organisms represent a unique setting for the elucidation of novel short proteins. Within this context, any Open Reading Frame (ORF) is potentially a protein-encoding gene. For prokaryotic genomes, the most accurate way to predict a gene is via similarity to a protein in another genome. This technique is problematic, however, due to limited numbers of experimentally verified short proteins in sequence databases. Further complicating the problem is a contamination of sequence databases caused by the propagation of dubious ORF predictions via homology-based annotation efforts [9].

In situations where there is no matching protein, sequence-composition-based methods are traditionally used. There are several automated genefinders that fall into this category such as GLIMMER [19, 6], ORPHEUS [7], GeneMark [13, 1, 2], and EasyGene [12, 15]. GLIMMER uses interpolated Markov models to distinguish coding from non-coding DNA. The program combines first through eighth-order Markov models and weights them by their predictive power. ORPHEUS begins by a database similarity search. The genes with matches in the database are then used to create a statistical profile of protein coding regions and ribosome binding sites. This profile is then used to make genome-wide predictions for protein-coding genes. The original GeneMark program used non-homogeneous Markov models to distinguish coding from non-coding sequences. The newer GeneMark-hmm program embeds the original GeneMark models into a hidden Markov model framework. EasyGene uses sequences that match a protein in Swiss-Prot to estimate an HMM for a given genome. The HMM is then used to score putative genes.

Prokaryotic-specific gene prediction algorithms have also been previously described. The Multivariate Entropy Distance (MED) algorithm, combines a comprehensive statistical model of protein coding ORFs with a model of prokaryotic Translation Initiation Sites (TISs). A novel feature of this algorithm is that the statistical model is based on a linguistic “Entropy Distance Profile” (EDP) which is inferred from observed amino acid probabilities. This profile is then used to map a given sequence within a 20-dimensional EDP phase space. Coding and non-coding sequences are then discriminated in part by how they cluster within this 20-dimensional space. Prior to this work, Edward Ochman investigated the ability to identify short bacterial proteins via a method that measured the ratio of nucleotide substitution rates between non-synonymous and synonymous mutation sites [16]. This method is based on a prior observation that among a set of closely related protein coding sequences, divergence at synonymous sites is greater than at non-synonymous sites [22, 11]. Ochman’s study only looked at previously annotated ORFs in bacterial genomes. All the gene finders described above work on single genomes, taking little advantage of conservation signals available with comparative genomics.

Short proteins represent a particularly difficult problem for all methods using sequence composition. Given that the ORFs are small, sequence composition analysis yields weak statistics, making it hard to discriminate a protein-encoding ORF from an ORF occurring due to chance. Genome annotators are often left with a difficult descision: to predict or not to predict. Using a large minimum size for predictions reduces the false positives but yields severe under-annotation for short proteins. Conversely, lower minimum sizes lead to over-annotation of prokaryotic genome sequences for short putative ORFs. An analysis in 2001 by Skovgaard et al. estimated that as much as 10% of the original annotation for the *E. coli* genome published in 1997 was a result of over-annotation, particularly for short ORFs [23].

We have developed a method to discriminate true short protein coding ORFs from an ensemble of all possible un-annotated ORFs in a given genome. Our method combines traditional gene annotation techniques with a novel application of local protein structure prediction tools within a comparative genomic framework. The novel assumptions in our method are that ORFs encoding short proteins will be selected for structural stability of the protein and for high frequency codons in a given genome. Just as for larger proteins, we hypothesize that structural stability will be critical to protein function. Therefore, we should be able to recognize conservation for properties related to structural stability in multi-genome alignments of closely related organisms. We validated our method on the complete genome sequence of *E. coli* strain K12 MG1655. Approximately 60 short proteins have been annotated and experimentally validated for this organism, representing the largest repertoire of validated short proteins described thus far [9]. Somewhat surprisingly, a large proportion of these proteins have been found to be associated with membranes.

2 Methods

2.1 Overview

In order to determine whether a given sequence codes for a short protein, we begin with a multiple genome alignment for our target organism. We then use this alignment to generate scores for each sequence based on three categories of analysis. The first analysis we perform is to analyze the observed codon composition for a given sequence according to a log-odds score. We score each sequence for agreement with its genome's known codon biases. Second, we analyze each sequence for protein-like conservation patterns in the multiple sequence alignment. We score an alignment of a homologous sequence to the target sequence according to a BLOSUM90 substitution matrix. We then compare this to the score of the target sequence aligned to itself. We expect homologous sequences that code for proteins to have a score similar to the target self-alignment score. Finally, we look for prediction strength and consistency among a set of local structure alphabets. For each sequence we generate three independent predictions for a given local structure alphabet and measure their overall agreement. We hypothesize that sequences coding for a protein will generate more consistent predictions than sequences not coding for a protein. We combine these scores, for a set of positive and negative training examples, to generate a model which we use to predict on new sequences.

2.2 Data compilation

To take advantage of an existing wealth of prokaryotic comparative genomic data and analysis tools at UC Santa Cruz, we obtained all sequence data and gene annotation directly from the UCSC Microbial Genome Browser (http://microbes.ucsc.edu/) [20]. For this analysis we generated a set of all possible open reading frames,10 amino acids or longer, in the entire genome for *E. coli* K12 . This set was then filtered to remove sequences with more than 20% overlap to any annotated genes in GenBank. We then removed any overlap to an experimentally verified set of 60 short proteins, previously described by Storz et al [9] so that we could use them for validating the method. The final set of ORFs contained 12,514 sequences. For a representative set of true protein encoding genes, we chose all genes annotated in GenBank as protein coding that were 1000 bases or longer. This list consisted of 1,625 sequences. We also looked at all ncRNAs annotated in GenBank. This set consisted of 168 sequences.

2.3 Multiple alignment generation

To make predictions for *E. coli* K12, we started with a multiple-genome alignment including 15 unique *E. coli* strains and 7 other Enterobacteriaceae: *Blochmannia floridanus, Buchnera aphidicola, Enterobacter 638, Salmonella enterica ATCC 9150, Salmonella enterica CT18, Shigella flexneri,* and *Yersinia pestis*. These genomes were selected on the basis of their relationship to *E. coli* K12. We wanted to include both closely and more distantly related species in our analysis as long as the genomes could be easily aligned. The multiple genome alignment file (MA) used in this study was created using the program Threaded Blockset Aligner (TBA) [3]. A phylogenetic tree derived from an analysis of 23S rRNA was used as one of the inputs for TBA. One of the advantages of using TBA is the ability to make any organism in a multiple alignment the reference genome. This ensures a 1:1 alignment mapping for all regions in the reference genome. Genome duplication events are not explicitly handled by TBA. For our study, *E. coli* K12 was the reference genome.

All sequences that were perfectly conserved in the multiple alignment were omitted from further analysis. The lack of any mutations inhibited measurement of protein-like conservation, thus removing a critical component of our study. This filter reduced the number of ORFs analyzed to 11,244, 1,585, 164, and 59 for all non-annotated ORFs, GenBank annotated ORFs greater than 1000 bases, annotated ncRNAs, and the experimentally validated set of short proteins respectively.

2.4 Codon bias calculations

For each genome, we made two models of codon probabilities: one based on observed counts in all GenBank-annotated protein genes for that genome, and the other on the GC-richness of the genome (provided by the genome browser). We then made a log-odds scoring system for each codon c,

$$\log_2 \frac{P_{\text{genome codon table}}(c)}{P_{\text{genome GC-richness}}(c)} ,$$

and averaged it over all codons in the ORF and over all aligned genomes. This codon-bias term measures selection for high expression and common amino acids in each genome, and turned out to be our strongest single predictor of protein-encoding ORFs. In order to investigate the impact of MA data on the method's performance, we also calculated the codon bias in the absence of any alignment data, using only the data for *E. coli* K12.

2.5 Amino acid conservation and BLOSUM-loss

To look for protein-like conservation, we converted the nucleotide alignments to amino-acid alignments. For each column in the multiple alignment we scored each pairwise target-sequence-to-homolog-sequence alignment according to a BLOSUM90 substitution matrix. We then computed a weighted average for this target-to-homolog score across all homologs in the alignment. To measure protein conservation in a given alignment column, we compared the weighted average for the target-to-homolog score to the BLOSUM90 score for the target sequence aligned to itself (1).

$$\frac{\sum_h W_h S_{ij}}{\sum_h W_h} / S_{ii} \tag{1}$$

where h = homologs in the alignment, S_{ij} = target-to-homolog BLOSUM90 score, S_{ii} = target-to-self BLOSUM90 score. Because we were specifically interested in mutations that are consistent with protein coding, we averaged these ratios over all codon positions that had at least one base different in at least one genome.

The homolog-specific weight W_h was set to 1 minus the computed sequence identity between the two genomes. Therefore, sequences that are very closely related had their scores down-weighted while scores from more distantly related sequences were given a higher weight (See Supplementary Material for a complete list of weights).

According to this scoring scheme then, positions that are perfectly conserved at the amino-acid level yield a ratio of 1, conservative mutations will decrease the ratio somewhat, and non-conservative mutations will decrease it substantially. To simplify plotting, we subtracted this average ratio from 1.01 to generate a BLOSUM-loss measure. A loss of 0.01 indicates perfect conservation at the protein level and large losses indicate non-protein-like mutations.

2.6 Local structure predictions

All local structure predictions were generated using Predict-2nd [10], using the amino-acid multiple alignments derived from the original DNA multiple alignment as inputs. We generated predictions for a set of 15 local structure alphabets, though only three of the fifteen ended up being used in our final predictor of protein-coding ORFs. To test the value of the comparative genomics input, we also made predictions using the same neural nets (NN), but with only the *E. coli* K12 translated ORFs as inputs, without the other genomes.

For each alphabet, we generated predictions from three independently trained neural nets. Each was trained on the same set of proteins, but using different multiple sequence alignments and different starting conditions for the optimization. The neural nets were not specially built for finding small proteins—they were available from the structure prediction work done for CASP8.

We then compared the output probability vectors for each target position to look for agreement among the three predictions (a, b, and c). To measure agreement we calculated the "dot product" of the resulting probability vectors for each position, $\sum_i a_i b_i c_i$, and took the average across the entire target sequence. This value is maximized when all three NNs generate strong predictions for a consistent local structure sequence—a signal we expected to be indicative of protein-like peptides.

2.7 Three-fold cross training

We performed three-fold cross validation experiments using a logistic regression model implemented in R [17] and two sets of training data. The negative training set was the set of ORFs in *E. coli* K12 with no more than 20% overlap to an existing GenBank annotation. The positive training set was the set of GenBank-annotated *E. coli* protein genes 1000 bases or longer. Each training set was split into three parts and two parts were used to train a logistic repression model, which was then tested on the remaining part. The training and test was repeated three times, once for each held-out third of the data.

To determine what combination of features to use in the logistic regression model, we used a simple greedy algorithm. We started by looking at the performance of all 17 features (15 local structure alphabet agreement scores, 1 codon bias score, and 1 BLOSUM-loss score) independently of one another. Then we select the best performing single feature where performance was determined by how many false positives were produced at a threshold that accepted half the real protein ORFs as true positives. We then repeated the analysis with all possible pairs of features containing the best individual score, then took the best performing pair of features and looked at all possible three feature combinations containing this pair. We repeated this process, selecting the best 2-, 3-, 4-, 5-, 6-, and 7-feature combinations. We did not go beyond 7 features because performance saturated at this point—in fact, we had to look at other thresholds besides TP=half the proteins ORFs in order to continue the greedy algorithm past 5 features, as no further changes at that threshold were visible when adding a 6th feature.

2.8 Validation Run

Because our model selection method used all the training data to help select the models, it does not directly tell us what to expect when applied to new data. Also, we defined all short ORFs as negatives for training purposes, but we are trying to find short ORFs that do code for proteins. We used the 5 features selected for the best 5-feature model in cross validation to make a predictor for validation with the experimentally validated set of small proteins.

The actual predictions were made by taking the average of 20 logistic regression models. Each model was trained on a data set containing 1000 positive and 1000 negative examples randomly selected from the training data. After the training, the models were used to predict all training data not used in building the model as well as the 60 experimentally validated short protein sequences (which had been excluded from both the negative and the positive training data). As a negative control, we used the same 20 regression models to make predictions for all 168 GenBank-annotated ncRNAs.

3 Results

3.1 Distributions of individual scoring features

We plotted the distribution of scores for positive and negative examples for all of our features (See Supplementary Material), and looked at how well each feature discriminated the two training populations. It was clear that by all measures the set of true protein coding sequences had a very distinguishable distribution. This information was very useful during the development and optimization of our scoring features.

3.2 Cross training results

The results from the systematic analysis of all 17 features was that the codon bias calculation was the best single feature. We generated true-positive-vs-false-positive curves for each feature. At 800 TPs, roughly half of positive training examples, the codon bias calculation has only 85 FPs (Table 1). The next best single feature was the BLOSUM-loss score with 425 FPs after the first 800 TPs, a significant drop-off in performance. Figure 2 shows the TP vs. FP curves for the cross-validation tests of the 7 models. We can clearly see the improved performance as we move from one to two, three, four and even five features. After this point, however, we seem to have saturated our performance. The best 7-feature combination included codon bias, CB-burial-14-7, CB8-sep9, BLOSUM-loss, n-notor2, Bystroff, and n-notor, added in that order. Figure 1 shows the curves for each individual component of the best 7-feature model. Table 2 shows the coefficients for each independent feature in the optimal 5-feature logistic regression model, the five features used for the validation test.

Alphabet	NUM FP
Codon bias	85
BLOSUM-loss	425
str4	674
Strand-sep	678
Alpha	714
Bystroff	881
CB-burial-14-7	1005
str2	1051
Protein blocks	1287
n-notor2	2121
n-notor	2207
Near-backbone-11	2684
n-sep	2795
CB8-sep9	3098
o-notor	3197
o-notor2	3632
o-sep	4162

Table 1: Table of the number of false positives at 800 true positives for each single-feature logistic regression model. All 17 features are listed in order of performance on the cross-validation test.

Alphabet	Coefficient
Codon bias	10.92752
CB-burial-14-7	-105.2917
CB8-sep9	235.8210
BLOSUM-loss	-5.071879
n-notor2	4.028408

Table 2: Regression coefficients for each of the parameters in the optimal 5-feature logistic regression model. Higher values indicate more emphasis placed on a particular feature. Positive values indicate a direct correlation between outcome and a given feature. Conversely, negative values indicate an inverse relationship between outcome and a given feature. The negative value was expected for BLOSUM-loss, which is minimized for true proteins, but was somewhat surprising for the the local structure alphabet CB-burial-14-7 (See Supplementary Material).

3.3 Impact of multiple alignment data

We were curious how well we could do if we did not have any comparative genomic data, as would be the case for a newly-sequenced genome that is not closely related to other bacterial genomes. We

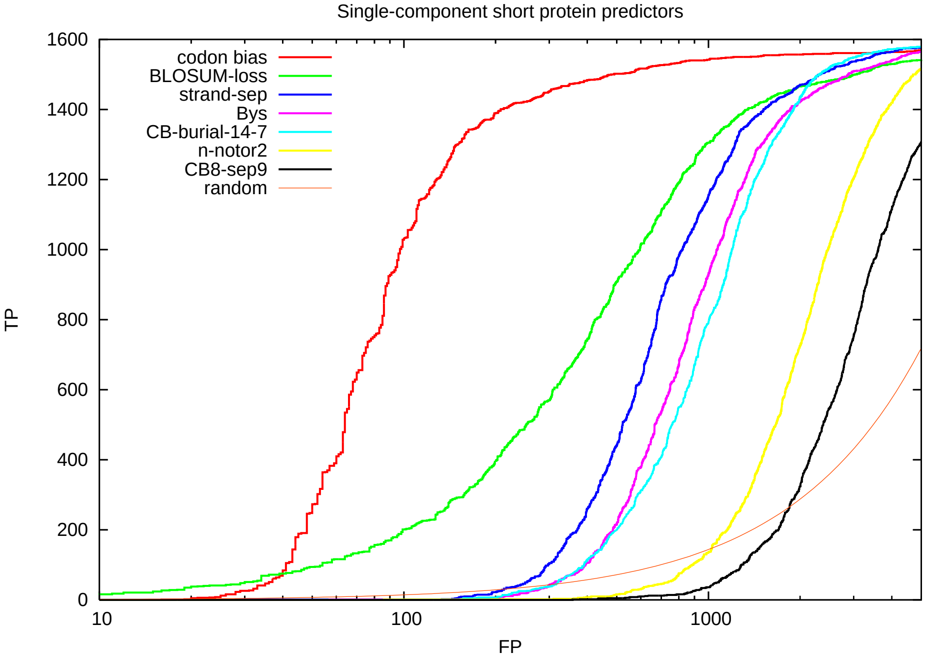

Figure 1: Average cross-validation results for single feature logistic regression models. These are all of the metrics included in the optimal 7-feature logistic regression model. The codon-bias score is clearly the best performing single scoring feature. Note that the order the feature was added does not correspond to their order as single feature. For example, the BLOSUM-loss measure is the second best here but carries much of the sane information as codon bias, and so is the 4th feature added, not the second.

know that our neural-net predictors are less accurate when given only single sequences as inputs, rather than alignments, so we expected a considerable drop in performance. Figure 3 shows the cross-validation results for using just the codon bias, and for the best logistic regression model with and without the multiple alignment data. Surprisingly, in the absence of any multiple alignment data, we were able to train a seven feature logistic regression model (CB-burial-14-7, single-genome codon bias score, near-backbone 11, protein blocks, n-notor 2, o-notor, o-notor2) that performed on par, in terms of the cross validation tests, with the best five-feature model built from multiple alignment data.

3.4 Validation

The validation experiment showed a surprisingly large number of the experimentally validated set of short proteins among the top prediction ranks. Using the best 5-feature combination of data (codon bias, CB-burial-14-7, CB8-sep9, BLOSUM-loss, n-notor2) we observed 4, 18, and 24 out of the top 10, 50, and 100 respectively were from the experimentally validated set of short proteins. We found half (30) of all the true short proteins within the top 200 predictions out of 11,467.

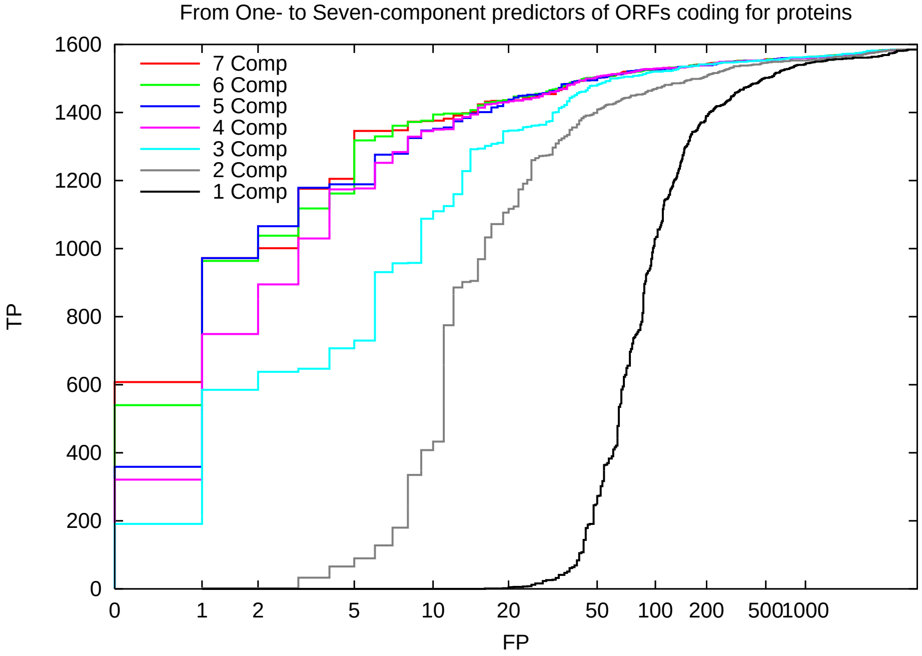

Figure 2: Average cross-validation results for 1-feature through 7-feature logistic regression models. Features were added in the order codon bias, CB-burial-14-7, CB8-sep9, BLOSUM-loss, n-notor2, Bystroff, n-notor. Performance improves only slightly after 5 features have been added.

We also weighed how much impact the multiple alignment data had on the validation experiment. Using only the single-genome-based codon bias data as a single-feature model did not get us any of the true short proteins within the top 200 predictions. However, if we included multiple alignment data into the codon bias score, the single-feature model was able to identify 14 of the validated short proteins among the top 200. Interestingly, the best combination of features generated in the absence of any multiple alignment data was able to find only 12 out of 60 true short proteins in the top 200. Overall, the alignment data resulted in a substantial improvement to performance in the validation experiment.

3.5 ncRNAs

The GenBank-annotated ncRNAs produced both expected and not-so-expected results. Overall, the set of annotated ncRNAs were correctly identified as not protein coding by the 5-feature predictor using multiple alignment data—the median rank for all ncRNAs was 3484. However, 7 annotated ncRNAs ranked among the top 200 predictions, see Table 3. Five of the seven (b4603, b4451, b4434, b1574, and b4439) were annotated as being in a class of regulatory RNA species that modulate expression of a target gene by complementary base pairing to the target RNA. Therefore, the observed strong predictions for these sequences may be explained by their complementarity to a true protein coding sequence. This species of regulatory RNAs are also known for signature

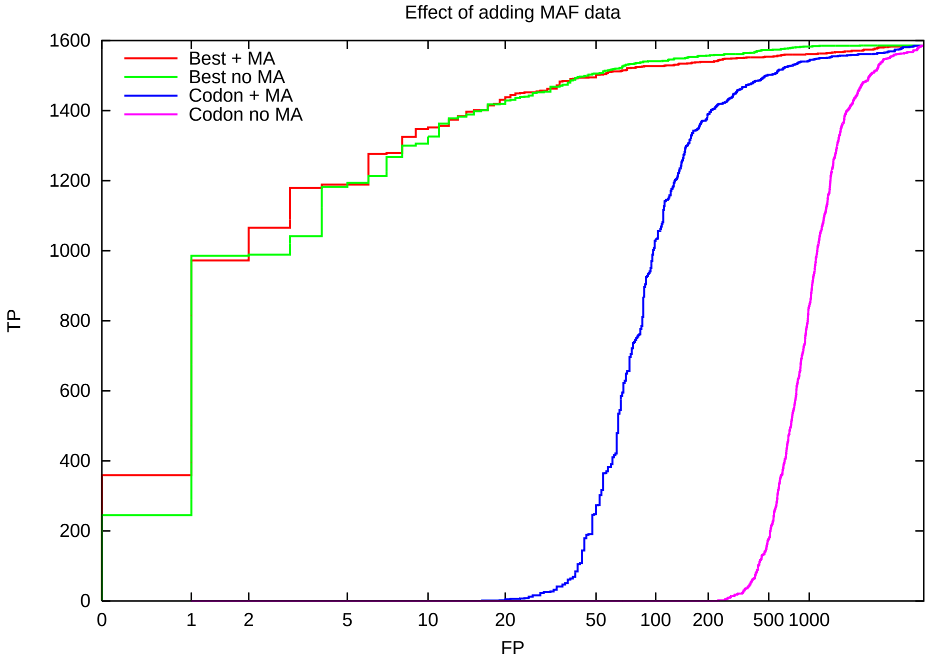

Figure 3: The multiple alignment (MA) data improves the codon-bias feature as a predictor enormously, but by adding more features, we can get cross-training results with a single genome that are almost as good as the comparative genomics results. The best-no-MA model used 7 features: CB-burial-14-7, single-genome codon bias, near-backbone-11, protein blocks, n-notor2, o-notor, and o-notor2, while the best-MA model used 5 features:codon bias, CB-burial-14-7, CB8-sep9, BLOSUM-loss, and n-notor2. Note that the BLOSUM-loss feature is not available in the single-genome case.

ID	Annotated Function	Probability
b4603	sRNA	0.98996
b4451	sRNA	0.96593
b4434	sRNA	0.91933
b1574	sRNA	0.75449
b1032	Ser tRNA	0.73255
b4439	sRNA	0.70371
b0883	Ser tRNA	0.67118

Table 3: GenBank-annotated ncRNAs appearing within top 200 predictions for the 5-feature predictor using multiple alignment data. The average “probability” (average output from the 20 regression models) for all GenBank-annotated ncRNAs was 0.08.

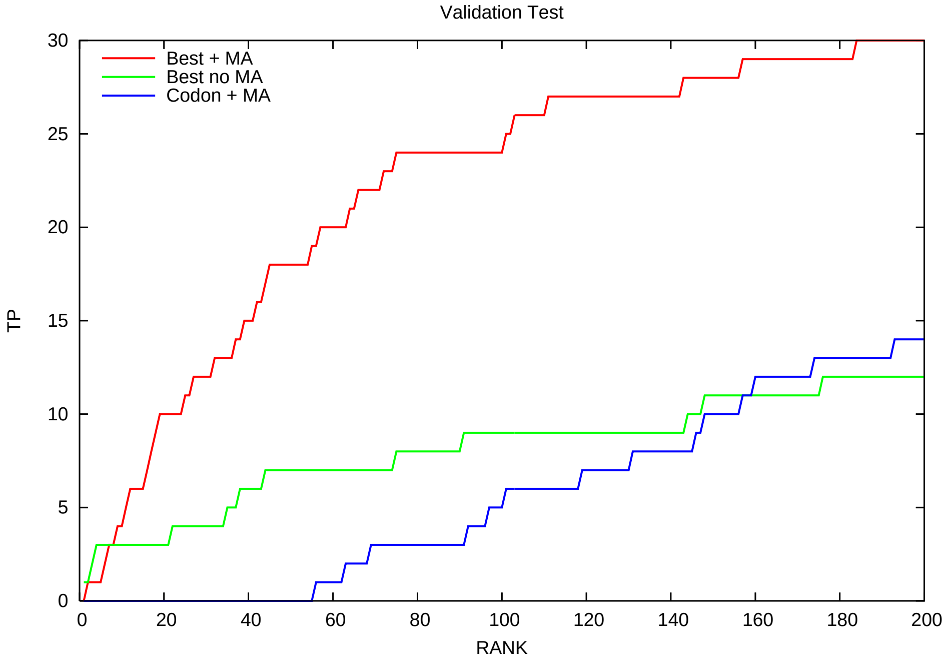

Figure 4: Validation results on experimentally verified short proteins. The use of multiple alignment (MA) data improved the performance substantially, but even without multiple alignment data there are 12 experimentally validated short ORFs in the top 200 hits (out of 11,467 ORFs scored). Using the best single feature, codon-bias, we found 14 out of the top 200 predictions were experimentally verified. However, when we calculated the codon bias score without multiple alignment data, we did not find any of the experimentally validated short proteins in the top 200 hits (Data no shown).

secondary structures which are essential to their activity. We expect to be able to separate such sRNAs from short protein ORFs by looking for complementary sequence in known protein genes, expected secondary structure, and the presence or lack of a predicted ribosome binding site.

4 Discussion

We have developed a method that uses traditional measures, protein conservation and codon bias, as well as local protein structure properties to generate predictions that a short ORF encodes for a protein. The framework for all our calculations is comparative genomics using whole genome alignments between closely related species. We have shown that within this context, multiple alignment information can be very valuable.

As we expected, sequence composition and protein-like conservation alone do not perform as well as our combined approach. Specifically, we were unable to identify any of the validated short proteins within the top 200 predictions when we used a single-genome codon bias metric as our lone data source. This is especially interesting because this approach is one of the standard "off-the-shelf" tools used to distinguish protein coding sequences. While this approach may work for longer sequences, our results show that short proteins will be missed by such efforts. The protein conservation score was clearly one of the best single features. However, it was the 4th metric added during our feature selection analysis, suggesting that the codon bias score captured most of the information in the protein conservation signal.

We were also seeing an enrichment for a specific class of ncRNAs in our top prediction ranks. We found five very highly predicted sRNAs within the top 200 predictions. It may be that these RNA species represent a significant source of false positives for our method. In order to avoid such contamination we are in the process of incorporating additional signals to our prediction strategy. Specifically, we are interested in predicted ribosome binding sites, complementary sequences among known genes, whole genome transcription profiles and other transcription analysis tools. These additional layers of information should improve our prediction accuracy. Furthermore, if our method is in fact enriching for regulatory sRNAs then an additional utility for our method could be the identification of novel regulatory ncRNAs.

For our analysis of the *E coli* K12 genome, we intentionally omitted all ribosome binding site data as a possible scoring feature. In several bacterial genomes, there are a large number of leaderless protein genes, so we created a method that did not rely on strong ribosome binding sites. Furthermore, a majority of the experimentally validated short proteins in this genome were identified in large part due to a strong ribosome binding site prediction near the ORF start site. Therefore, inclusion of this information would have given us a less stringent test of our method. Recovering the known short proteins without using this signal further demonstrates the validity of our approach.

Our method has two main criteria that are required in order to generate a set of predictions. First is a multiple genome alignment of closely related species containing the organism of interest. Note however, that this is a soft requirement as single-genome predictions are possible though of substantially lower quality. The second requirement is a set of positive and negative sequence examples in order to train a logistic regression model. This does not mean however, that we require a comprehensive annotation set for a given genome. Minimally, we can use the set of all ORFs longer than 1000 bases as a positive training set. These sequences should be highly enriched for true protein coding sequences. Conversely, a negative training set could be built from all possible ORFs below an arbitrary size cutoff, say 30 amino acids, the large majority of which should be non-protein coding.

As presently constructed our method is highly adaptable to new genome sequences. We are in the process of adding our method to the UCSC Microbial Browser creation pipeline. This should enable us to produce predictions for any completed genome in the public databases.

We are extremely encouraged by the performance of our method in-silico. However, we know that the final validation of our predictions must come from experimental techniques. Therefore, we are currently designing experiments to validate high confidence predictions in other bacterial genomes, specifically the human pathogenic species *Vibrio cholerae.*

References

[1] J Besemer and M Borodovsky. Heuristic approach to deriving models for gene finding. *Nucleic Acids Res*, 27(19):3911–3920, Oct 1999.

[2] J Besemer, A Lomsadze, and M Borodovsky. GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Res*, 29(12):2607–2618, Jun 2001.

[3] M. Blanchette, W.J. Kent, C. Riemer, L. Elnitski, A.F. Smit, K.M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E.D. Green, D. Haussler, and W. Miller. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.*, 14:708–715, Apr 2004.

[4] N. Cloonan and S. M. Grimmond. Transcriptome content and dynamics at single-nucleotide resolution. *Genome Biol.*, 9:234, 2008.

[5] S. Cutting, M. Anderson, E. Lysenko, A. Page, T. Tomoyasu, K. Tatematsu, T. Tatsuta, L. Kroos, and T. Ogura. SpoVM, a small protein essential to development in Bacillus subtilis, interacts with the ATP-dependent protease FtsH. *J. Bacteriol.*, 179:5534–5542, Sep 1997.

[6] A L Delcher, D Harmon, S Kasif, O White, and S L Salzberg. Improved microbial gene identification with GLIMMER. *Nucleic Acids Res*, 27(23):4636–4641, Dec 1999.

[7] D Frishman, A Mironov, H W Mewes, and M Gelfand. Combining diverse evidence for gene recognition in completely sequenced bacterial genomes. *Nucleic Acids Res*, 26(12):2941–2947, Jun 1998.

[8] R. L. Gallo and V. Nizet. Endogenous production of antimicrobial peptides in innate immunity and human disease. *Curr Allergy Asthma Rep*, 3:402–409, Sep 2003.

[9] M. R. Hemm, B. J. Paul, T. D. Schneider, G. Storz, and K. E. Rudd. Small membrane proteins found by comparative genomics and ribosome binding site models. *Mol. Microbiol.*, 70:1487–1501, Dec 2008.

[10] S. Katzman, C. Barrett, G. Thiltgen, R. Karchin, and K. Karplus. PREDICT-2ND: a tool for generalized protein local structure prediction. *Bioinformatics*, 24:2453–2459, Nov 2008.

[11] S. Kumar and S. R. Gadagkar. Efficiency of the neighbor-joining method in reconstructing deep and shallow evolutionary relationships in large phylogenies. *J. Mol. Evol.*, 51:544–553, Dec 2000.

[12] Thomas Schou Larsen and Anders Krogh. EasyGene–a prokaryotic gene finder that ranks ORFs by statistical significance. *BMC Bioinformatics*, 4:21, Jun 2003.

[13] A V Lukashin and M Borodovsky. GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res*, 26(4):1107–1115, Feb 1998.

[14] U. Nagalakshmi, Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, and M. Snyder. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*, 320:1344–1349, Jun 2008.

[15] Pernille Nielsen and Anders Krogh. Large-scale prokaryotic gene prediction and comparison to genome annotation. *Bioinformatics*, 21(24):4322–4329, Dec 2005.

[16] H. Ochman. Distinguishing the ORFs from the ELFs: short bacterial genes and the annotation of genomes. *Trends Genet.*, 18:335–337, Jul 2002.

[17] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[18] M. V. Ruvolo, K. E. Mach, and W. F. Burkholder. Proteolysis of the replication checkpoint protein Sda is necessary for the efficient initiation of sporulation after transient replication stress in Bacillus subtilis. *Mol. Microbiol.*, 60:1490–1508, Jun 2006.

[19] S L Salzberg, A L Delcher, S Kasif, and O White. Microbial gene identification using interpolated Markov models. *Nucleic Acids Res*, 26(2):544–548, Jan 1998.

[20] K. L. Schneider, K. S. Pollard, R. Baertsch, A. Pohl, and T. M. Lowe. The UCSC Archaeal Genome Browser. *Nucleic Acids Res.*, 34:D407–410, Jan 2006.

[21] B. A. Sela. [Hepcidin–the discovery of a small protein with a pivotal role in iron homeostasis]. *Harefuah*, 147:261–266, Mar 2008.

[22] L. C. Shimmin, P. Mai, and W. H. Li. Sequences and evolution of human and squirrel monkey blue opsin genes. *J. Mol. Evol.*, 44:378–382, Apr 1997.

[23] M. Skovgaard, L.J. Jensen, S. Brunak, D. Ussery, and A. Krogh. On the total number of genes and their length distribution in complete microbial genomes. *Trends Genet.*, 17:425–428, Aug 2001.

[24] C. Yanofsky. Transcription attenuation: once viewed as a novel regulatory strategy. *J. Bacteriol.*, 182:1–8, Jan 2000.