# A Probabilistic Genome-Wide Gene Reading Frame Sequence Model

Christian Theil Have[1] and Søren Mørk[2]

[1] Novo Nordisk Foundation Center for Basic Metabolic Resarch, Section of Metabolic Genetics, Copenhagen University, c.have@sund.ku.dk
[2] Center for non-coding RNA in Technology and Health, Copenhagen University, soer@rth.dk

**Abstract.** We introduce a new type of probabilistic sequence model, that model the sequential composition of reading frames of genes in a genome. Our approach extends gene finders with a model of the sequential composition of genes at the genome-level – effectively producing a sequential genome annotation as output. The model can be used to obtain the most probable genome annotation based on a combination of i: a gene finder score of each gene candidate and ii: the sequence of the reading frames of gene candidates through a genome. The model — as well as a higher order variant — is developed and tested using the probabilistic logic programming language and machine learning system PRISM - a fast and efficient model prototyping environment, using bacterial gene finding performance as a benchmark of signal strength. The model is used to prune a set of gene predictions from an underlying gene finder and are evaluated by the effect on prediction performance. Since bacterial gene finding to a large extent is a solved problem it forms an ideal proving ground for evaluating the explicit modeling of larger scale gene sequence composition of genomes.

We conclude that the sequential composition of gene reading frames is a consistent signal present in bacterial genomes and that it can be effectively modeled with probabilistic sequence models.

## 1 Introduction

Automated genome annotation is essential for exploiting the enormous amounts of genome sequence data currently being generated [2]. The initial steps of genome annotation relies heavily on probabilistic nucleotide sequence models, for generating sets of predicted genes. Such models typically estimate the probability that each open reading frame (ORF) is a gene. This estimate is usually based on only a limited context comprising the ORF nucleotide sequence and perhaps a few hundred bases upstream and downstream to include signals such as promoters and ribosomal binding sites. The subsequent steps to assemble a genome annotation typically involves selecting the highly scoring predictions using a significance criteria or threshold. In some recent gene finders [5, 7, 6] the selection of predictions is done as a genome-wide optimization where the predictions are chosen to form a coherent genome annotation by taking into account the extent of overlap between genes.

In a similar vein, we introduce a probabilistic sequence model which select the set of predictions that form the genome annotation, but which is based on sequential composition of gene reading frames, which we believe is a novel signal to be explored in gene finding. Our purpose is not to build the next state-of-the-art gene finder, but to present a class of simple models which clearly demonstrates the efficacy of exploiting the gene-reading-frame-sequence bias.

The existence of a gene-strand bias in prokaryotes is well established [3]. One source for this bias is a tendency for genes to be placed on the leading strand due to replication efficiency consequences of co-directional and head-on collisions of the replication and transcription apparatus [10]. It has also been argued that the preferential placement of genes in the leading strand is driven by essentiality rather than expression [12].

A gene-reading-frame-sequence bias is a general signal that can incorporate gene-strand bias, bias due to clusters of orthologous genes [13], operonic structures [16], phase preference for overlapping genes [4] and other potential effects yielding non-random sequence composition.

The gene-strand bias account for a large proportion of the gene-reading-frame-sequence bias, but a pronounced bias is detectable even within the strands. Furthermore, the gene-reading-frame-sequence bias seems to be symmetric for the two strands, cf. Table 1. This is a convenient property, especially considering the arbitrary designation of which is the forward and which is the reverse strand.

## 2 Methods

Our gene-reading-frame-sequence model are implemented in PRISM, a probabilistic logic programming language and machine learning system with generic algorithms for parameter estimation and decoding [14]. We use PRISM as a convenient model comparison platform, since it is powerful enough to express the different models and enables a level execution provided by its generic machine learning routines. The use of probabilistic logic programming for evaluating sequence models as the heart of contemporary gene finders has recently been demonstrated in [9].

The gene-reading-frame-sequence model — which we call *Frameseq* – is a variant of a fully connected Hidden Markov Model (HMM) [11] with a state for each of the six possible reading frames — the *frame states* — and a delete state. Given a sequence of gene predictions sorted by position, a path through the model capable of emitting this prediction sequence represents a classification of predictions into presumed true positives emitted from the frame states and presumed false positives emitted from the delete state. A path with optimal probability represents a best hypothesis about the classification of predictions into positives and negatives. This path can be calculated using the Viterbi algorithm which is provided by PRISM.

Each state emits a score symbol and a frame for each gene prediction. Frame states only emit predictions with a corresponding frame, whereas the delete state may emit predictions of any frame. The score symbol is a symbolic value representing a range of confidence scores for the predictions of the input gene finder. The emission probabilities thus reflect the prediction confidence scores in the training set.

Traditionally, the transition probabilities of an HMM are conditioned only on the previous state (the Markov property). In our model the transition probability is conditioned on the previous *frame state* rather than just the previous state. The frame state transition probabilities are thus assumed to reflect the probability of a seeing a gene in a particular reading frame given the reading frame of the previous gene.

Higher ordered Markov models have generally shown to be an improvement over standard models for the nucleotide sequence models used in bacterial gene finding (*e.g.* as used in Genemark and Glimmer). To explore the possibility that the same might be true for the gene reading frame sequence, we have also employed a second order version of Frameseq, *i.e.*, which conditions transitions on the two previous frame states.

The transition probabilities between the *frame states* are estimated as the relative frequency of observed adjacent genes in the various frames observed in the set of verified genes.

The probability of a transition to the delete state (from any state) reflects the probability that a gene finder prediction is a false positive,

$$P(delete) = 1 - \frac{TP}{TP + FP}$$

where $TP$ is the number of true positives predicted by the gene finder and $FP$ is the number of false positives. This probability is directly related to gene finder specificity and may be tweaked for different sensitivity/specificity trade-offs. We exploit this in experiments reported below.

The frame state transition probabilities are estimated as relative frequencies, which have the interpretation of conditional probabilities given that a transition to the delete state did not occur. We normalize each of these transition probabilities by multiplying them by $1 - P(delete)$.

Each state is capable of emitting a finite set of $i$ symbols $\delta_1 \ldots \delta_n$ corresponding to ranges of prediction scores, $i.e.$n the states emit a discretized symbol corresponding to the confidence score of a prediction as supplied by the gene finder. The ranges are selected to ensure that each score symbol correspond to an equal proportion of gene finder predictions. The number of ranges, $n$, is a configurable parameter; when $n$ is high the model can better exploit the scores from the gene finder, but the estimated emission probabilities become more fragile, *i.e.*, more data is needed to reliably estimate them. The emission probabilities of the delete state are estimated as the relative frequency of each of the possible score symbols for all false positives predictions, *i.e.*,

$$P(\delta_i | state = delete) = \frac{FP_{\delta_i}}{FP}$$

where $FP_{\delta_i}$ is the number of false positives with a confidence score within the range symbolized by $\delta_i$.

Similarly, the emission probabilities of frame states are estimated as the fraction of true positive predictions belonging to a particular range within the corresponding frame, *i.e.*,

$$P(\delta_i|state = frame_j) = \frac{TP_{\delta_i}^{frame_j}}{TP^{frame_j}}$$

where $TP^{frame_j}$ is the total number true positive predictions in reading frame $j$ and $TP_{\delta_i}^{frame_j}$ is the number of true positive predictions in reading frame $j$ with a confidence score within the range symbolized by $\delta_i$.

A illustration of the states and transitions and of the model is shown in figure 1.

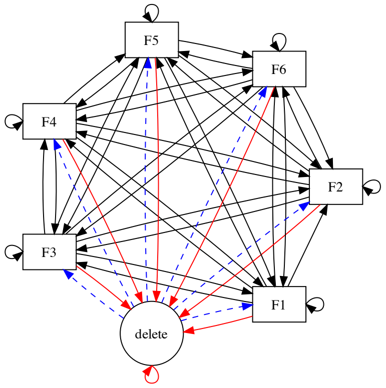

**Fig. 1.** The Frameseq delete-HMM model. All frame states F1 . . . F6 have transitions to each other and to themselves. Transitions to the delete state are symbolized by red arrows, to indicate that they share the transition probability, $P(delete)$. The dashed blue arrows illustrate transitions from the delete state to a frame state – the probability of which depend on the last frame state visited before the delete state. Furthermore, the delete state is drawn as circle rather than a box to convey that it resembles a silent state – it does produce emissions (predicted false positives) but we are only interested in emissions from the frame states (predicted true positives). To minimize visual clutter, a begin and end state have been omitted.

Instead of using exact empirical frequency counts as described above, we use a variational Bayes version of the EM algorithm [15] provided with PRISM. This algorithm puts Dirichlet priors (pseudo-counts) on random variables ensuring that all estimated probabilities are non-zero.

# 3 Results and Discussion

## 3.1 The Phylogenetic Reach of the Gene Reading Frame Bias

To test the generalization capability and potential phylogenetic reach of our model, we train models on five different prokaryotic genomes and use them to filter predictions for the *E. coli* genome. We expect *E. coli* to have the most reliable genome annotation and by using it for validation we obtain the most reliable validation results. By training on distant organisms, we show the robustness of our approach with regard to both training set quality and phylogenetic distance. Good performance on *E. coli* should also imply that we can train our model on a well annotated genome and filter gene finder predictions in other genomes with increased reliability. To validate this we also do this experiment in reverse, *i.e.*, we also train our model on *E. coli* to predict on each of the other genomes. For all models trained, we set the number of score ranges to $n = 15$.

The five genomes, listed here in ascending order of phylogenetic distance from *E. coli*: *Escherichia coli* [REFSEQ:NC_000913], *Salmonella enterica* [REFSEQ:NC_004631.1], *Legionella pneumophila* [REFSEQ:NC_002942], *Bacillus subtilis* [REFSEQ:NC_000964] and *Thermoplasma acidophilum* [REFSEQ:NC_002578].

We use Genemark 2.5 [1] which is available as a web-service to produce a large initial set of candidate genes. Genemark is currently available in newer versions (GeneMarkS and GenemarkHMM) with improved prediction performance. However, as our main interest is to introduce a new type of genome-sequence model, not to improve gene finding, the older version of Genemark provides a number of advantages for our purposes that are not present in other available single-sequence gene finders: Genemark 2.5 use a very simple scoring model and do not employ any post-scoring prediction selection algorithm, but is capable of producing a large set of predictions simply by enforcing a (low) score cut-off. As it does not otherwise prune predictions, we eliminate factors which could affect and reduce the pruning potential available to our model. Obviously, the accuracy of this gene finder is slightly below what is now state-of-the-art.

The full dataset offered by being able to produce a large set of predictions provides a better evaluation of the contribution of the reading frame signal than pruning a small optimal prediction set or (for completeness we do include such more limited experiments for state-of-the-art gene finders below).

We set the configurable score cut-off as low as possible, *i.e.*, to 0.1, to allow as many false positive predictions as possible. The gene finder predictions are preprocessed to contain only the best scoring prediction for each distinct stop codon. For each genome, we train using the preprocessed Genemark predictions and use the RefSeq annotation as golden standard. By inspection of transition probabilities, we observe that the gene-reading-frame-sequence bias tends to be almost symmetric for the strands, see table 1.

We test the performance of each model on the Genemark predictions for the target genome by measuring sensitivity and specificity in terms of predicted stop codons with respect to the RefSeq annotation.

**Table 1.** Estimated transition probabilities between frame states. A cell indicates the probability that a gene in the frame indicated by the row is followed by the gene in the frame indicated by the column. Note that the strands have almost symmetrical probabilities.

| from \ to | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.18 | 0.20 | 0.28 | 0.13 | 0.1 | 0.1 |
| 2 | 0.29 | 0.2 | 0.2 | 0.12 | 0.09 | 0.1 |
| 3 | 0.22 | 0.3 | 0.17 | 0.1 | 0.11 | 0.1 |
| 4 | 0.11 | 0.1 | 0.1 | 0.19 | 0.23 | 0.27 |
| 5 | 0.11 | 0.9 | 0.1 | 0.29 | 0.19 | 0.22 |
| 6 | 0.09 | 0.08 | 0.1 | 0.23 | 0.30 | 0.19 |

*E. Coli*

| from \ to | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.19 | 0.23 | 0.30 | 0.09 | 0.09 | 0.09 |
| 2 | 0.29 | 0.20 | 0.25 | 0.08 | 0.10 | 0.07 |
| 3 | 0.23 | 0.29 | 0.18 | 0.09 | 0.11 | 0.10 |
| 4 | 0.10 | 0.09 | 0.10 | 0.21 | 0.22 | 0.28 |
| 5 | 0.11 | 0.10 | 0.10 | 0.28 | 0.18 | 0.22 |
| 6 | 0.11 | 0.10 | 0.10 | 0.20 | 0.30 | 0.21 |

*S. Entirica*

| from \ to | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.18 | 0.23 | 0.26 | 0.10 | 0.11 | 0.13 |
| 2 | 0.27 | 0.18 | 0.22 | 0.12 | 0.09 | 0.12 |
| 3 | 0.24 | 0.27 | 0.18 | 0.10 | 0.12 | 0.10 |
| 4 | 0.09 | 0.11 | 0.11 | 0.18 | 0.20 | 0.31 |
| 5 | 0.12 | 0.11 | 0.11 | 0.26 | 0.17 | 0.23 |
| 6 | 0.09 | 0.09 | 0.11 | 0.24 | 0.29 | 0.18 |

*L. pneumophila*

| from \ to | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.22 | 0.24 | 0.25 | 0.09 | 0.10 | 0.10 |
| 2 | 0.29 | 0.20 | 0.23 | 0.11 | 0.08 | 0.09 |
| 3 | 0.25 | 0.25 | 0.22 | 0.08 | 0.10 | 0.11 |
| 4 | 0.09 | 0.10 | 0.06 | 0.23 | 0.24 | 0.28 |
| 5 | 0.11 | 0.09 | 0.09 | 0.26 | 0.22 | 0.23 |
| 6 | 0.10 | 0.07 | 0.08 | 0.26 | 0.28 | 0.21 |

*B. subtilis*

| from \ to | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.18 | 0.20 | 0.23 | 0.17 | 0.09 | 0.13 |
| 2 | 0.25 | 0.16 | 0.24 | 0.13 | 0.13 | 0.09 |
| 3 | 0.22 | 0.26 | 0.21 | 0.11 | 0.10 | 0.10 |
| 4 | 0.12 | 0.14 | 0.12 | 0.18 | 0.23 | 0.20 |
| 5 | 0.15 | 0.16 | 0.06 | 0.24 | 0.18 | 0.20 |
| 6 | 0.10 | 0.11 | 0.15 | 0.20 | 0.22 | 0.21 |

*T. acidophilum*

We repeat this process with incrementally increasing delete state probabilities resulting in a range of sensitivity/specificity trade-offs. These are plotted in Figure 2 as a Receiver Operator Characteristic (ROC) curves.

For comparison we provide a baseline ROC curve, produced via incrementally increasing a cut off value of the scores for the Genemark predictions for the target genome.

For all organisms except the phylogenetically very distant *T. acidophilum*, Frameseq improves accuracy and the margin of the improvement correlates with phylogenetic distance. The pronounced improvement in the accuracy which can be observed in ROC curves for the frame-bias model as compared to the baseline demonstrates that for comparable levels sensitivity, Frameseq achieves a lower false positive rate.

### 3.2 A Higher Order Signal?

It is plausible that the gene-reading-frame-sequence bias is more complex than just pairwise dependencies between the frames of genes. More complex depen-

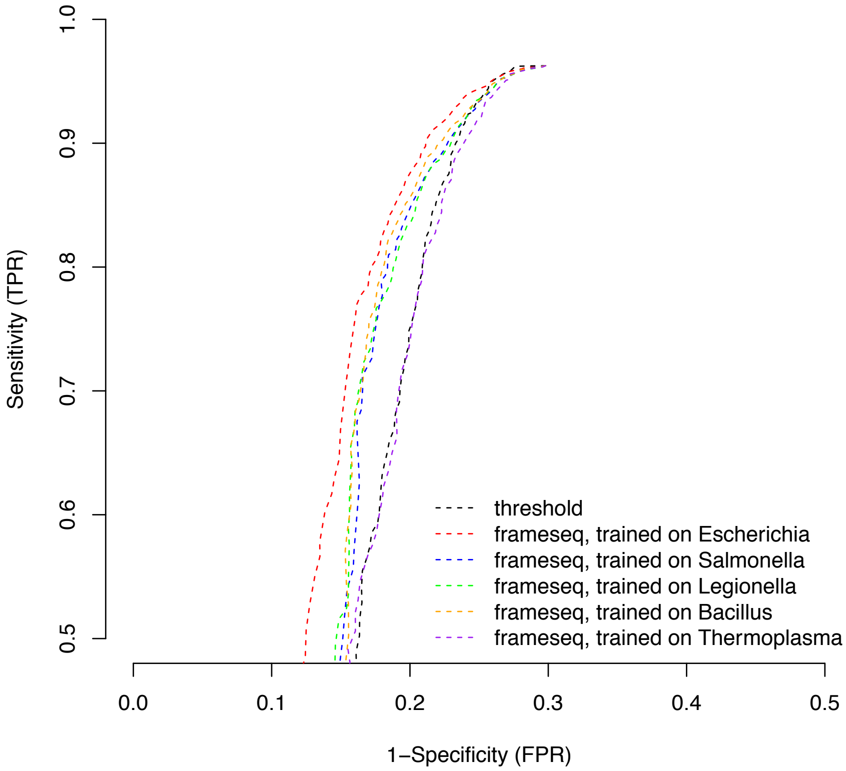

**Fig. 2.** Phylogenetic reach. The figure shows ROC curves for filtering of all Genemark 2.5 predictions with score $> 0.1$ for different organisms. The black curve shows selection using a threshold and the colored curves show filtering using Frameseq. The Frameseq model is trained the *E. coli* for all organisms. The experiment shows that it is possible to train Frameseq on a well-known and well-annotated organism and apply it to filter predictions on phylogenetically distant organisms with improved accuracy. Accuracy is improved for both *S. enterica* and *B. subtilis* and in part for *L. pneumophila*, but not for the phylogenetically distant *T. acidophilum*.

dencies on previous gene reading frames can be modeled using a higher order model.

We test this hypothesis by using a second order HMM based Frameseq model which is trained and applied on *E. coli*. We compare this to the basic Frameseq model which uses a first order HMM. We also investigate the phylogenic of conservation of a possible higher order signal by training the same model on *S. enterica* and decoding on *E. coli*. In both cases, the we use predictions from the Genemark 2.5 gene finder, with a score cut-off of 0.1. As in the previous experiments we set the number of score ranges to $n = 15$.

We derive and compare ROC curves for threshold selection and Frameseq selection like in the previous experiments, but here for both the first order and second order models (Figure 3).

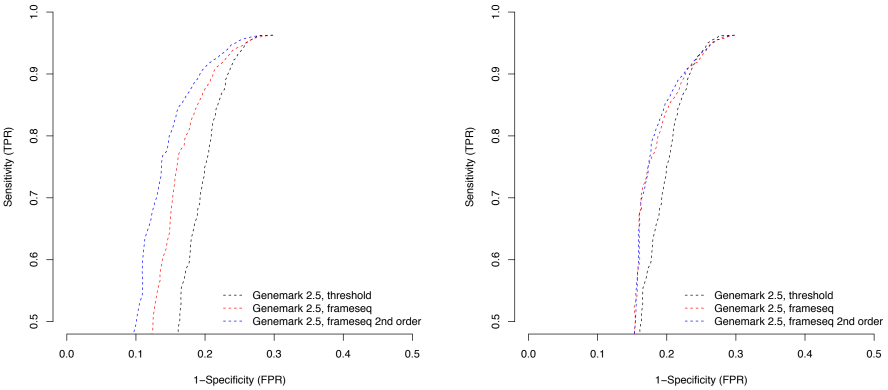

**Fig. 3.** The ROC curves compares the basic first order Frameseq model to a second order model. The black curve indicate threshold selection, the red curve is the first order model and the blue curve is the second order model. In the panel to the left the models are trained and used on *E. coli*and in panel to the right they are trained on *S. enterica*and used on *E. coli*. The second order model results in When trained and applied on the same genome, the second order achieves markedly better accuracy than with the first order model. However, when trained on genome and applied to another genome the second order model achieves only marginally better accuracy compared to the first order model.

In the case where we train on *E. coli*, the second order Frameseq model results in significantly better accuracy than with the first order model. The improvement degrades quite a bit when we instead train the model on *S. enterica*, but there is still a detectable improvement in accuracy for the second order model.

It should be noted that, higher order models effectively increase the amount of transition probabilities involved, but the amount of training data used to estimate these are fixed in our case. This means that increasing the order of the model results in less reliable transition probabilities. This may explain some of the loss of accuracy when observed when training on *S. enterica* as compared to training on *E. coli.*

On the other hand, the experiment with the second order Frameseq model trained on *E. coli* demonstrates the maximal potential of utilizing a higher order signal.

### 3.3 Effect on State-of-the-art Gene Finders

In this section we explore using Frameseq with Glimmer 3 and Prodigal 2.50 — to evaluate the contribution of a reading frame sequence signal for state-of-the art gene finders.

Genemark 2.5 which was used in the previous experiments, scores each open reading frame individually and does not attempt to stitch such individual predictions together into a more coherent set of predictions for the genome.

The algorithms employed by the two other gene finders have some similarities to the delete-HMM of Frameseq. Both gene finders use custom dynamic programming algorithms to achieve a more coherent set of predictions for a genome. Prodigal use several features including hexamer scores, ribosomal binding site detection, maximal overlap and distance between predictions combined using a custom dynamic programming algorithm. Similarly, Glimmer 3 also use a dynamic programming algorithm which restricts the size of overlaps between predictions. Neither of these algorithms utilize the gene frame bias.

Our algorithm is quite simplistic in comparison since it only considers one signal – the gene-reading-frame-sequence bias. It could undoubtedly be improved by considering other signals and constraints inherent between predictions such as distance and overlaps. In being simplistic, however, it clearly demonstrates the utility of the gene-reading-frame-sequence bias without the inherent noise from the impact of other features – which is our purpose.

In these experiments, we use Frameseq to filter the predictions of the state-of-the-art gene finders in order to explore the potential beneficial effect they could achieve by incorporating the gene-reading-frame-sequence bias. For each gene finder — Prodigal 2.50 and Glimmer 3 — we apply the second order Frameseq model trained on *E. coli* to filter their respective predictions on also on *E. coli.* The number of score ranges is in this experiment set to $n = 100$ to better capture the more detailed variations of the scores. The results are shown in figure 4.

In all cases the filtered predictions have significantly improved specificity for comparable levels of sensitivity. The effect of Frameseq seems most pronounced with reduced sensitivities which could indicate that the scores of the gene finders are more reliable for the top-scoring predictions.

These experiments do not conclusively prove that all the gene finders could achieve improved specificity for the desired levels of sensitivity (close to one) by incorporating the gene-reading-frame-sequence bias. It should be noted that we slightly over-fit the model by training on *E. coli* and by doing this we get more impressive results than would have been the case if the models where trained

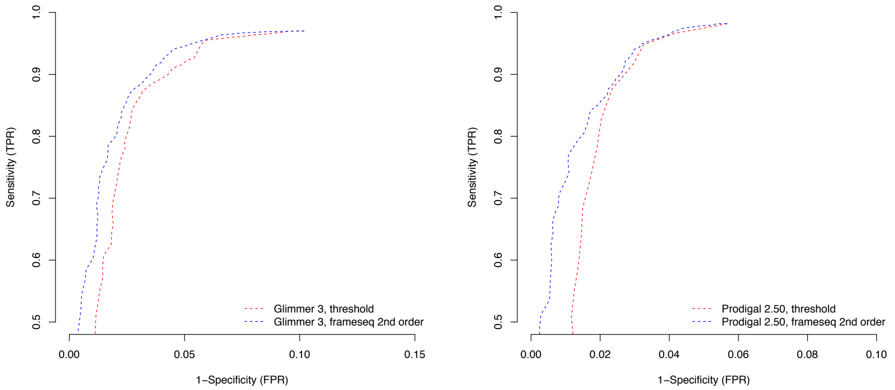

**Fig. 4.** Left: Frameseq with Glimmer. The red ROC curve shows threshold selection with Glimmer 3 predictions on *E. coli* and blue curve shows results of filtering these predictions with a second order Frameseq model trained on *E. coli.* Right: Frameseq with Prodigal: The red ROC curve shows threshold selection with Prodigal 2.50 predictions on *E. coli* and blue curve shows results of filtering these predictions with a second order Frameseq model trained on *E. coli.*

using other organisms. Training Frameseq on, *e.g.*, *S. enterica* and filtering predictions for these gene finders does not result in significantly improved accuracy (data not shown). We believe this to be mainly a problem of sparsity of the training data, but also due to the reduced margins for possible improvement as compared to Genemark 2.5. We demonstrated the phylogenetic reach using Genemark 2.5, but the margin for possible improvement is significantly smaller with Glimmer and Prodigal. Due to this, a slightly under-fitted model will generalize sufficiently to improve Genemark 2.5 results, but insufficiently with the state-of-the art gene finders.

The experiment here, however, does show that the gene-reading-frame-sequence bias signal provides useful information which is complementary to the signals used by the contemporary methods.

## 4 Conclusions

We have demonstrated the feasibility of modeling the sequential composition of genes in a genome with simple sequential reading frame models. We obtain surprisingly good results when predicting on one organism with models trained on phylogenetically distant genomes, which implies both the generality of the approach and the potential importance of gene reading frame sequence structure across taxa.

The impact of our method is most pronounced for reduced levels of sensitivity. Ideally we would like to achieve as significant improvements in specificity for a higher level of sensitivity, but improved specificity with a lower sensitivity is still a good result with important implications; It means that our approach is capable of supplying a larger set of gene predictions with a specified upper bound on the false positive rate, than is possible with any other gene finder. This may be useful when selecting candidate genes for experimental verification and can reduce the likelihood of wasted lab effort.

We also believe that the gene-reading-frame-sequence bias signal can be useful for improving automated computational genome annotation, but in order to achieve this, it will need to be integrated with the algorithms of state-of-the-art gene finders instead of the relatively superficial augmentation we do here.

In order to clearly illustrate the gene-reading-frame-sequence bias, we engineered our method to be as simple as possible, which in effect have several limitations:

- It relies on gene finder scores rather integrating with the algorithm of the gene finder, thereby missing out on exploiting possible correlations with signals incorporated in the gene finder.
- It relies on discretization of gene finder scores, *i.e.*, it summarizes of the information contained in prediction scores and hence cannot fully exploit these. The discretization procedure could be improved by using variable sized bins, *e.g.*, as in [8], or by instead using a continuous Hidden Markov Model.
- We do not fully exploit the nature of the gene finder score distribution for parameter smoothing. We do apply limited parameter smoothing by using the variational Bayes EM algorithm, but we could probably achieve better generalization by fitting a suitable function to the gene finder score distribution.
- We use only a single organism as training data which become too sparse; This results in slightly over-fitted models when training on the same genome and slightly under-fitted models when training on an other genome. This situation could be amended by training on several genomes.

Despite these limitations, our method achieves good results which illustrate the usefulness of the signal, yet still leaves room for potential improvement.

We choose the problem of bacterial gene finding to exemplify the gene-reading-frame-sequence bias and its use. This problem has the nice property that it is almost solved, which enables us to use reference annotations to validate the approach. It should be noted, however, that many reference annotations are unverified results from the gene finders that we try to improve upon. This bias gives our method a slight disadvantage.

Lastly, we believe that the gene-reading-frame-sequence bias signal could have applications beyond gene finding. For instance, it may potentially benefit next generation sequencing and genome assembly where a complete model of the overall gene content of a genome would be applicable.

*Availability* The source code of the model and accessory scripts are freely available at: http://github.com/frameseq/frameseq

*Acknowledgement* This work is part of the project “Logic-statistic modeling and analysis of biological sequence data” funded by the NABIIT program under the Danish Strategic Research Council.

## References

1. J. Besemer and M. Borodovsky. Genemark: web software for gene finding in prokaryotes, eukaryotes and viruses. *Nucleic Acids Res.*, 33:451–454, 2005.
2. M. R. Brent. Steady progress and recent breakthroughs in the accuracy of automated genome annotation. *Nature Reviews Genetics*, 9(1):62–73, 2008.
3. B. J. Brewer. When polymerases collide: replication and the transcriptional organization of the e. coli chromosome. *Cell*, 53:679–686, 1988.
4. P. J. A. Cock and D. E. Whitworth. Evolution of relative reading frame bias in unidirectional prokaryotic gene overlaps. *Molecular Biology and Evolution*, 27(4):753–756, 2010.
5. A. L. Delcher, K. A. Bratke, E. C. Powers, and S. L. Salzberg. Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics*, 23:673–679, 2007.
6. C. T. Have. Constraints and global optimization for gene prediction overlap resolution. In A. D. Palu, A. Dovier, and A. Formisano, editors, *Proceedings of Workshop on Constraint Based Methods for Bioinformatics*, 2011.
7. D. Hyatt, G. Chen, P. LoCascio, M. Land, F. Larimer, and L. Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC bioinformatics*, 11(1):119, 2010.
8. P. Kontkanen and P. Myllymäki. MDL histogram density estimation. *Journal of Machine Learning Research - Proceedings Track*, 2:219–226, 2007.
9. S. Mørk and I. Holmes. Evaluating bacterial gene-finding hmm structures as probabilistic logic programs. *Bioinformatics*, 28(5):636–642, 2012.
10. R. T. Pomerantz and M. O’Donnell. The replisome uses mRNA as a primer after colliding with RNA polymerase. *Nature*, 456:762–766, 2008.
11. L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, volume 77, 1989.
12. E. P. C. Rocha and A. Danchin. Essentiality, not expressiveness, drives gene-strand bias in bacteria. *Nature genetics*, 34:377–378, 2003.
13. I. Rogozin, K. Makarova, J. Murvai, E. Czabarka, Y. Wolf, R. Tatusov, L. Szekely, and E. Koonin. Connected gene neighborhoods in prokaryotic genomes. *Nucleic Acids Research*, 30(10):2212–2223, 2002.
14. T. Sato. Generative Modeling by PRISM. *Proceedings of the International Conference on Logic Programming*, LNCS 5649:24–35, 2009.
15. T. Sato, Y. Kameya, and K. Kurihara. Variational bayes via propositionalized probability computation in prism. *Annals of Mathematics and Artificial Intelligence*, 54(1-3):135–158, Nov. 2009.
16. Y. Wolf, I. Rogozin, A. Kondrashov, and E. Koonin. Genome alignment, evolution of prokaryotic genome organization, and prediction of gene function using genomic context. *Genome research*, 11(3):356–372, 2001.