A first step towards learning which uORFs regulate gene expression

Bryant, CH, Kemp, GJL and Cvijovic, M

http://dx.doi.org/10.2390/biecoll-jib-2006-31

A First Step towards Learning which uORFs Regulate Gene Expression

Selpi[1], Christopher H. Bryant[1], Graham J.L. Kemp[2], Marija Cvijovic[3]

[1]School of Computing, The Robert Gordon University, St. Andrew Street, Aberdeen, AB25 1HG, United Kingdom, {selpi,chb}@comp.rgu.ac.uk

[2]Department of Computer Science and Engineering, Chalmers University of Technology, SE-412 96, Göteborg, Sweden, kemp@cs.chalmers.se

[3]Max Planck Institute for Molecular Genetics, Department Lehrach, Kinetic Modelling Group, Ihnestrasse 63-73, 14195, Germany, cvijovic@molgen.mpg.de

Summary

We have taken a first step towards learning which upstream Open Reading Frames (uORFs) regulate gene expression (i.e., which uORFs are functional) in the yeast *Saccharomyces cerevisiae*. We do this by integrating data from several resources and combining a bioinformatics tool, ORF Finder, with a machine learning technique, inductive logic programming (ILP). Here, we report the challenge of using ILP as part of this integrative system, in order to automatically generate a model that identifies functional uORFs. Our method makes searching for novel functional uORFs more efficient than random sampling. An attempt has been made to predict novel functional uORFs using our method. Some preliminary evidence that our model may be biologically meaningful is presented.

1 Introduction

Regulation of gene expression is central to biology. However, a holistic regulatory mechanism of gene expression is still far beyond current knowledge in biology. This is mainly because very little is known about regulatory elements. In this research, we explore the possibility and challenges of combining a machine learning technique, inductive logic programming (ILP) [14], with a bioinformatics tool, ORF Finder [21] (`http://bioinformatics.org/sms/orf_find.html`), to data integrated from several data resources (Saccharomyces Genome Database, EMBL Database and the supplementary material of [18]) to learn about one of the regulatory elements, namely the upstream Open Reading Frames (uORFs), in the yeast *Saccharomyces cerevisiae*. To the best of our knowledge, this is the first time that such a combination has been applied to this particular domain.

Given a set of uORFs which regulate gene expression, the learning task for ILP is to automatically generate a model (a set of rules) which can then be used to predict whether unseen uORFs regulate gene expression. This task has become very important to biologists because it could lead to a deeper understanding of how uORFs are involved in the regulatory mechanism of gene expression. This learning task is very challenging because lab experiments to test whether a gene contains functional uORF(s) are costly and time consuming, and currently available data are incomplete and of poor quality [6].

Figure 1: From DNA to protein

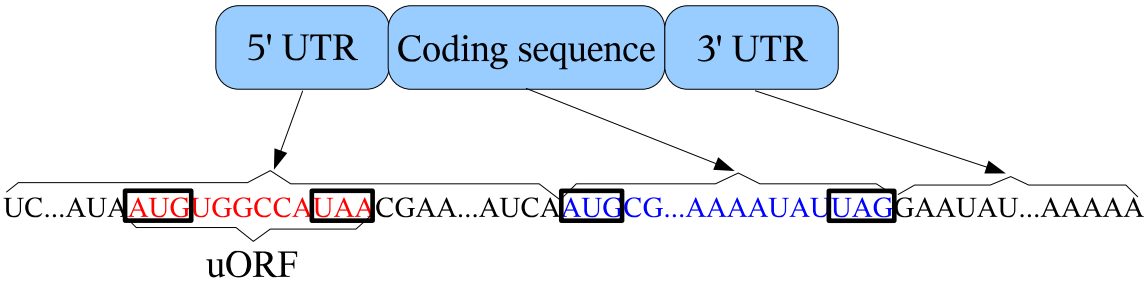

Figure 2: Simplification of mRNA structure. A, U, G, and C are RNA's bases. A codon is a triplet of bases. AUG is the start codon. A stop codon can be UAA, UAG, or UGA. A 5′ UTR may have zero or more uORFs.

A model organism such as *S. cerevisiae* makes it possible to design experiments to verify whether particular uORFs do indeed regulate translation. Our work is directed towards helping to select sets of candidate functional uORFs for such experimental studies.

This paper is structured as follows. Section 2 describes the biological background, motivation, and objectives of this study. Section 3 introduces the machine learning technique used, namely inductive logic programming. The data used in this study, and the way in which examples and background knowledge are represented, are introduced in Section 4. Section 5 describes the ILP training method used to generate a model for predicting functional uORFs. Section 6 describes the performance measure used. The results of using the ILP-generated model to predict novel functional uORFs in *S. cerevisiae* are presented in Section 7. Some related work is discussed in Section 8. Finally, Section 9 concludes the paper and discusses some future research directions. Supplementary material for the study presented in this paper is available at `http://www.comp.rgu.ac.uk/staff/chb/research/data_sets/jib2006/uORF/`.

2 Biological Background, Motivation, and Objectives

Deoxyribonucleic acid (DNA) carries a complete set of instructions for making all the proteins a living cell will ever need. A segment of DNA which contains the information for protein synthesis is called a gene. Transcription of DNA produces ribonucleic acid (RNA) molecules which will be used to produce proteins (see Figure 1).

One group of RNA molecules, called messenger RNA (mRNA), carries the instructions from DNA out of the nucleus into the cytoplasm for protein synthesis. mRNAs contain untranslated regions (UTR) at their 5′ and 3′ ends (see Figure 2). These UTRs, specifically the 5′ UTR, are known to play several key roles in post-transcriptional regulation of gene expression [24, 10, 20, 3, 26, 19]. However, it is not yet clear through what mechanism the UTRs regulate the translation process.

One of the regulatory elements that may be present in the 5′ UTR is the upstream Open Read-

ing Frame (uORF) [17]. A uORF is identified by the presence of both a start codon before (i.e. *upstream* of) the start codon of the main coding sequence, and an in-frame stop codon, as illustrated in Figure 2. Research has revealed that several transcribed uORFs regulate the translation process (i.e., they are *functional*) (e.g. [24, 25, 5, 3, 6]), while a few others do not (i.e., they are *non-functional*) [11]. To get a better understanding of how uORFs regulate the translation process, it is important to first identify which uORFs are functional.

We have collected a set of 51,904 crude uORFs from 5,602 genes of the yeast *S. cerevisiae*. We describe this set as crude because it consists of uORFs which can be transcribed within mRNAs and those which cannot; uORFs which can regulate gene expression will only be found among the transcribed ones. One approach to searching for functional uORFs would be to sample genes at random and test their uORFs in the laboratory. The most direct test to verify that the uORFs are transcribed and whether they are functional is by measuring the levels of mRNA and protein of the native gene in its proper chromosomal context [3]. Such experiments are costly and time-consuming (≈ 4 man-months per gene, Sunnerhagen, P. personal communication).

It has been suggested that no more than 10% of the yeast genes will have one or more functional uORFs [10] and each of these genes will on average have two functional uORFs (Sunnerhagen, P., personal communication). Thus, if one searched for functional uORFs by selecting genes at random from the set of 5,602 genes and testing them in the lab, then on average it would take ≈ 20 man-months to find a single functional uORF. Therefore, an automated learning method to recognise functional uORFs is essential to support experimental lab work aiming to discover and verify functional uORFs in a cost-effective way. To date, no such method is available.

The importance of functional uORFs to uncovering the regulatory mechanism of gene expression and the need for an automated learning method to recognise functional uORFs motivated this study. Our objectives are: to automatically generate a set of rules (a model) which identifies functional uORFs using inductive logic programming; and then to use the resulting model to predict novel functional uORFs.

3 Inductive Logic Programming

Inductive logic programming (ILP) [8] is the area of Artificial Intelligence which deals with the induction of hypothesised predicate definitions of a concept (such as functional uORFs). Unlike most ML techniques, ILP is able to bias inference to take into account expert knowledge, such as existing knowledge of biological structures and phenomena. Such knowledge is referred to as *background knowledge* in ILP. ILP algorithms take examples of the concept, together with potentially pertinent background knowledge about the concept, and construct a hypothesis which explains the examples in terms of the background knowledge.

The declarative representation of examples, background knowledge and the induced hypotheses in ILP can be easily translated to English. Consequently biologists can help with the selection and integration of appropriate background knowledge and the final dissemination of discoveries to the wider scientific community.

In ILP we can represent knowledge in either an intensional or extensional manner [8]. Knowledge is described extensionally by listing the descriptions of all of its instances. For example, the lengths of all the uORFs. However an extensional definition can be undesirable for a number

Table 1: Detailed composition of prediction made by ORF Finder [21]

Number of Genes		Transcribed uORFs			Other uORFs
		Functional	Non-functional	Unknown	(Not known if transcribed)
17 studied genes	8	20	-	-	269
	2	-	2	-	8
	7	-	-	8	103
5,585 other genes		-	-	-	51,494
5,602 genes		20	2	8	51,874

of reasons, including the fact that the number of instances can be large. An intensional description is more compact and often takes the form of rules. For example a rule which identifies the shortest uORF in a UTR (see Table 4).

4 Data and Knowledge Representation for ILP

The collection of 51,904 uORFs from 5,602 genes of the yeast *S. cerevisiae* were collected using ORF Finder [21] (`http://bioinformatics.org/sms/orf_find.html`). Because the length of $5'$ UTR are only known for a small number of genes (only 248 genes can be assigned unambiguously from European Molecular Biology Laboratory (EMBL) database), ORF Finder was used to search for Open Reading Frames (a series of triplets of bases, which starts with a start codon and ends with a stop codon) in the intergenic (between two genes) sequences of the yeast *S. cerevisiae*. The lengths of intergenic sequences are taken from the supplementary material of [18].

17 of these 5,602 genes have been well-studied and are documented to have uORFs transcribed within their mRNAs, as summarised in [2, 24] The detailed composition of our data is summarised in Table 1. Recently, Zhang and Dietrich [28] reported 15 new genes which contain uORFs transcribed within their mRNAs. However, we did not include their findings for our experiments, rather we used their findings for the purpose of analysing the results of our ILP experiments (see Section 7).

Since our goal is learning to recognise which uORFs regulate gene expression, we can consider this learning task to be a classification problem. Ideally a typical classification system in machine learning learns from a mixture of positive and negative examples. In this domain, positive examples would be uORFs that are transcribed and regulate gene expression (i.e., functional) and negative examples would be uORFs that are transcribed but do not regulate gene expression (i.e., non-functional). The uORF data from 5,585 genes (see Table 1) are all unclassified. Hence, for the training stage in this study, only the uORF data of the 17 studied genes were used.

As summarised in Table 1, among the uORF data of the 17 studied genes, 20 uORFs have been verified experimentally as functional. These were used as positive examples. [2, p. 32] summarised that there are only 2 uORFs from 2 genes which have been verified to be non-functional. Therefore, there were only 2 negative examples in our data set. The rest of the transcribed uORFs (8 uORFs) and all other uORFs (which are not known to be transcribed) for those 17 genes ($269 + 8 + 103 = 380$ uORFs) were used as randoms. Here randoms are data that are likely to be negative, although there is still a small probability that the data are

Table 2: Detailed uORF composition from 17 studied genes within the prediction made by ORF Finder [21]

Gene Name[a]	Systematic Name[a]	Transcribed uORFs	Other uORFs	Positive Examples	Negative Examples	Random Examples
CLN3	YAL040C	1	7	1	-	7
GCN4	YEL009C	4	15	4	-	15
HAP4	YKL109W	2	26	2	-	26
TIF4631	YGR162W	5	202	5	-	202
YAP1	YML007W	1	3	1	-	3
YAP2	YDR423C	2	-	2	-	-
HOL1	YNR055C	1	15	1	-	15
PET111	YMR257C	4	1	4	-	1
SCO1	YBR037C	1	4	-	1	4
CBS1	YDL069C	1	4	-	1	4
INO2	YDR123C	1	9	-	-	10
PPR1	YLR014C	1	2	-	-	3
URA1	YKL216W	1	13	-	-	14
LEU4	YNL104C	1	12	-	-	13
RCK1	YGL158W	2	49	-	-	51
DCD1	YHR144C	1	17	-	-	18
SCH9	YHR205W	1	1	-	-	2
17 Genes		30	380	20	2	388

[a]Names are taken from SGD (http://www.yeastgenome.org).

positive. The detailed uORF composition from 17 studied genes within the prediction made by ORF Finder is described in Table 2.

Given the characteristics of the data (i.e., the number of negative examples is too few compared to the positive examples, and there is an abundance of random examples), we explore learning from positive and random examples only. For that purpose, we used the positive-only setting [16] of CProgol [15] version 4.4 [13]. CProgol 4.4 is an inductive logic programming (ILP) system, which has been applied in another domain with these characteristics (e.g. [12]).

CProgol 4.4 was instructed to learn a predicate `has_functional_role/1` from a set of training examples. **Positive** examples were represented as ground unit clauses of the predicate `has_functional_role(X)`, where `X` is a uORF ID. A uORF ID is a composite of the systematic name of the gene to which the uORF belongs (e.g., those listed in second column of Table 2) and a uORF identifier (e.g., uORF1, uORF2, etc.). The set of positive examples was divided into two parts, with two thirds (14 uORFs) of the data set used for training and the remaining one third (6 uORFs) used for testing. The 388 **random** examples were also partitioned, with two thirds used for training and the remainder used for testing.

In addition to positive and random examples, the ILP system was provided with extensional and intensional background knowledge.

Extensional Background Knowledge. [24, 2] suggested several important features that can determine the impact of a uORF on post-transcriptional gene expression, such as: the distance of the uORF from the start of the coding sequence in bases; the sequence context (the frequency of AU and GC base-pairs) upstream of (before) the uORF's start codon and downstream of (after) the uORF's stop codon; and the length of the uORF in codons. 5′ UTR related properties, such as the number of uORFs predicted by ORF Finder in the intergenic sequence, the length of intergenic sequence, and the relationship between UTR and uORF were also included (see

Table 3: Predicates representing background knowledge of uORFs and UTRs. A set of ground unit clauses was generated for each predicate.

`uORF(X,Y,Z)` where X is a uORF ID, Y is the distance of X from the start of coding sequence, and Z is the length of X.
`utr(X,Y,Z)` where X is a UTR ID, Y is the number of uORF that X has, and Z is the intergenic sequence length between X and the previous gene.
`has_uORF(X,Y)` where X is a UTR ID and Y is the uORF ID of one of X's uORFs.
`belongs_to(X,Y)` where X is a uORF ID and Y is a UTR ID to which X belongs.
`context(X,Y,Z)` where X is a uORF ID, Y and Z are the frequencies of AU and GC within 20 bases downstream of X's stop codon.
`up_context(X,Y,Z)` where X is a uORF ID, Y and Z are the frequencies of AU and GC within 20 bases upstream of X's start codon.

Table 4: Intensional Background Knowledge[a]

```
has_shortest_dist_in_UTR(UORF):-
    uORF(UORF,ShortestDist,_), belongs_to(UORF,UTR),
    setof(Dist,(has_uORF(UTR,UORFX),uORF(UORFX,Dist,_)),List),
    List = [ShortestDist|_].
has_shortest_len_in_UTR(UORF):-
    uORF(UORF,_,ShortestLen), belongs_to(UORF,UTR),
    setof(Len,(has_uORF(UTR,UORFX),uORF(UORFX,_,Len)), List),
    List = [ShortestLen|_].
gcrich_down_up(UORF):-
    context(UORF,Au,Gc), Gc > Au,
    up_context(UORF,A,G), G > A.
gcrich_down_aurich_up(UORF):-
    context(UORF,Au,Gc),  Gc > Au,
    up_context(UORF,A,G), G < A.
aurich_down_up(UORF):-
    context(UORF,Au,Gc), Gc < Au,
    up_context(UORF,A,G), G < A.
gcrich_up_aurich_down(UORF):-
    context(UORF,Au,Gc), Gc < Au,
    up_context(UORF,A,G), G > A.
```

[a]CProgol's built-in predicate setof(X,P,L) produces a list L of objects X that satisfy P. L is ordered and duplicate items are eliminated.

Table 3).

Intensional Background Knowledge. The declarative rules shown in Table 4 capture concepts that are potentially useful for helping to identify functional uORFs, and therefore might be included in the hypotheses induced by the ILP system. We matched the verified functional uORFs from [24, 2] to the uORF data obtained using ORF Finder. From this, we observed that majority of the functional uORFs are the closest one to the main coding sequence. Therefore, we defined a rule that identifies whether a uORF is closer to the coding sequence than all others within the same gene. Verified functional uORFs are often very short, so one might be interested to identify the shortest uORF of each gene. [23, 4] suggest that the sequence context of a uORF's start and stop codons have an impact on translation. Therefore, we defined rules that examine the abundance of AU and GC base pairs immediately upstream and downstream of each uORF.

Table 5: Mode declarations for generating a model that identifies functional uORFs[a]

```
:- modeh(1,has_functional_role(+uORF))?
:- modeb(1,uORF(+uORF,-distancefromstart,-codonlength))?
:- modeb(1,belongs_to(+uORF,-utr))?
:- modeb(1,utr(+utr,-numberofuORF,-intergeniclength))?
:- modeb(1,+distancefromstart=< #int)?  :- modeb(1,+codonlength=< #int)?
:- modeb(1,+distancefromstart>= #int)?  :- modeb(1,+codonlength>= #int)?
:- modeb(1,+distancefromstart= #int)?   :- modeb(1,+codonlength= #int)?
:- modeb(1,+intergeniclength=< #int)?   :- modeb(1,+numberofuORF=< #int)?
:- modeb(1,+intergeniclength>= #int)?   :- modeb(1,+numberofuORF>= #int)?
:- modeb(1,+intergeniclength= #int)?    :- modeb(1,+numberofuORF= #int)?
:- modeb(1,has_shortest_dist_in_UTR(+uORF))?
:- modeb(1,has_shortest_len_in_UTR(+uORF))?
:- modeb(1,gcrich_down_up(+uORF))?
:- modeb(1,aurich_down_up(+uORF))?
:- modeb(1,gcrich_down_aurich_up(+uORF))?
:- modeb(1,gcrich_up_aurich_down(+uORF))?
```

[a]modeh describes the clauses to be used in the head of a hypothesis, and modeb describes the clauses to be used in the body of a hypothesis.

5 Generating a Model that Identifies Functional uORFs

We investigate whether ILP could automatically generate a model that identifies functional uORFs, and whether this model, when used as a filter, could be more efficient than random sampling. The training set consists of 14 positives and 259 randoms, and the test set consists of 6 positives and 129 randoms.

CProgol's parameters were set as follows: **posonly** is set to 'on', so that CProgol learns from positives and randoms only; **inflate** (gives a weighing to the data/predicate in general) is set to 4,200%; **c** (the maximum number of atoms in the body of the rules constructed) is set to 6; **nodes** (the maximum number of nodes explored during clause searching) is set to 7,000; and **r** (the maximum depth of resolutions allowed when proving) is set to 700.

We defined the hypothesis space for CProgol 4.4, so that it can construct a definition for the target predicate `has_functional_role/1`. This was done by giving mode declarations (see Table 5).

The types `uORF` and `utr` were declared by defining a set of ground unit clauses of the predicate `uORF(X)`, where `X` is a uORF ID; and a set of ground unit clauses of the predicate `utr(X)`, where `X` is a UTR ID. The types of `codonlength`, `distancefromstart`, `intergeniclength`, and `numberofuORF` were all defined as integer. Figure 3 shows the resulting model.

6 Measuring Model Performance using Relative Advantage

An independent test set was used to evaluate the model. The default performance measure in CProgol 4.4 is predictive accuracy. However, this measure gives a poor estimate when used in a domain where positives are rare, which is the case in this uORF domain. Therefore, we do

```
has_functional_role(A) :- uORF(A,B,C), B=<204.
has_functional_role(A) :- uORF(A,B,C), belongs_to(A,D), B=<409,
    C=<6, utr(D,E,F), F>=589.
```

English translation: A uORF has functional role if it satisfies at least one of the following rules.

- if its distance from the start of coding sequence is less than or equal to 204;
- if its distance from the start of coding sequence is less than or equal to 409, its length is less than or equal to 6, and the intergenic length is greater than or equal to 589.

Figure 3: The model which predicts functional uORFs

Table 6: A summary of classification and performance measurement of experiment generating a model which predicts functional uORFs (in Section 5)

Positives correctly classified as positives	3
Randoms falsely classified as positives	4
Positives falsely classified as randoms	3
Randoms correctly classified as randoms	125
mean RA	17.3

not use this performance measure.

Instead we adapted Relative Advantage (RA) [12, Appendix A]. This uORF domain has the characteristics for which RA is claimed be useful. These include the fact that the proportion of positives (functional uORFs) in the example set is very small, while the proportion of positive examples in the population (the whole *S. cerevisiae* yeast genome) is not known, acquiring negatives is difficult (as this has to be verified via lab experiments), and a benchmark recognition method does not exist.

The idea behind using RA is to predict cost reduction in finding functional uORFs using the model compared to using random sampling. In this application domain, RA is defined as

$$RA = \frac{A}{B}; \text{ where}$$

A = the expected cost of finding a functional uORF by repeated independent random sampling from a set of 51,904 crude uORFs and testing each uORF in the lab.

B = the expected cost of finding a functional uORF by repeated independent random sampling from a set of 51,904 crude uORFs and analysing only those which are predicted by the model to be functional.

A summary of the classifications made and the performance measurement from the experiment in Section 5 is presented in Table 6. Using our model as a predictor makes the search for novel functional uORFs 17 times more efficient than random sampling. Reducing the number of randoms that are falsely classified as positives is very important in this domain, because verification via lab analysis is costly.

```
has_functional_role(A) :- uORF(A,B,C), belongs_to(A,D), B=<204,
    utr(D,E,F), E>=207.
has_functional_role(A) :- uORF(A,B,C), belongs_to(A,D), B=<409,
    C=<6, utr(D,E,F), E>=5.
has_functional_role(A) :- belongs_to(A,B), utr(B,C,589).
has_functional_role(A) :- uORF(A,B,C), has_shortest_dist_in_UTR(A),
    C=<8, B>=23.
has_functional_role(A) :- uORF(A,57,B).
has_functional_role(A) :- uORF(A,250,B).
```

English translation: A uORF has functional role if it satisfies at least one of the following rules.

- if its distance from the start of coding sequence is less than or equal to 204 and the UTR to which it belongs has at least 207 uORFs;
- if its distance from the start of coding sequence is less than or equal to 409, its maximum length is 6 codons, and the UTR to which it belongs has at least 5 uORFs;
- if intergenic length of its UTR to which it belongs is 589;
- if it is the closest uORF to the coding sequence within its UTR, its length is less than or equal to 8 codons, and its distance from the start of coding sequence is greater or equal to 23;
- if its distance from the start of coding sequence is 57;
- if its distance from the start of coding sequence is 250.

Figure 4: The model generated from the experiment to predict novel functional uORFs

7 Predicting Novel Functional uORFs

Although our model (Figure 3) looks simple, its mean RA value shows that the model makes the search for novel functional uORFs more efficient. Thus, it is expected that the positive-only setting of CProgol 4.4 can help in predicting novel functional uORFs. To support this argument, an experiment was conducted to predict novel functional uORFs. The method used was the same as that described in Section 5 except that the training set consists of 20 positives and 388 randoms from 17 studied genes. The resulting model was then used to predict novel functional uORFs from 51,494 randoms (from 5,585 genes, see Table 1 on page 4). Figure 4 shows the model generated from the experiment to predict novel functional uORFs. 5,595 out of 51,494 uORFs are predicted as functional uORFs by this model.

Clearly, extensive lab work would be required to verify whether these uORFs, which are predicted as functional by our model, are indeed functional. However, some promising indications are given by comparing our predictions with experimental lab results from a recent study by Zhang and Dietrich [28]. Further to the 17 genes and 30 verified transcribed uORFs mentioned in Table 1, Zhang and Dietrich [28] have reported an additional 15 genes which contain 19 verified transcribed uORFs in the yeast *S. cerevisiae*. Their focus was to find additional genes which contain transcribed uORF(s). Thus it is not clear which of these 19 newly verified transcribed uORFs are functional. However, as uORFs which can regulate gene expression are among the transcribed ones, we used their findings for the purpose of analysing the results of our ILP experiments.

Zhang and Dietrich [28] provide some evidence that our rules may be biologically meaningful. In their paper, they wrote "We observed that uORFs are present in over 95% of 250 bp 5′ upstream regions of *S. cerevisiae*". But for their analysis, a 210 bp (base pair) 5′ upstream re-

gion was used as the upper boundary to eliminate "spurious potential uORFs". This suggested that functional uORFs are likely to be found within 250 bp from the start of coding sequence (because the functional uORFs have to be transcribed). Our rules reflect that condition. Of the 15 genes reported by Zhang and Dietrich [28], our model predicts that 12 will have functional uORFs, and that 13 of the 19 transcribed uORFs will be functional (Table 7).

Table 7: Predictions made using the model in Figure 4 for the 15 genes reported by Zhang and Dietrich [28].

Gene Name	Systematic Name	uORF's Position	uORF's Length	uORF Identifier in [28]	uORF Identifier in this study[c]	Predicted as Functional
ARV1	YLR242C	-125	12	uORF1	uORF5	No
		-108	3	uORF2	uORF4	Yes
		-40	7	uORF3	uORF6	Yes
ECM7[a]	YLR443W	-15	5	uORF	-	-
HEM3	YDL205C	-129	9	uORF	uORF8	No
RPC11	YDR045C	-60	4	uORF	uORF5	Yes
AVT2	YEL064C	-11	4	uORF	uORF7	Yes
TPK1	YJL164C	-42	5	uORF	uORF4	Yes
MBR1	YKL093W	-70	7	uORF	uORF5	Yes
APC2	YLR127C	-27	5	uORF	uORF3	Yes
SPE4	YLR146C	-41	6	uORF	uORF5	Yes
SPH1	YLR313C	-25	4	uORF	uORF3	Yes
IMD4[a,b]	YML056C	-99	14	uORF	-	-
SLM2	YNL047C	-110	24	uORF1	uORF11	No
		-84	6	uORF2	uORF9	Yes
		-70	4	uORF3	uORF10	Yes
FOL1	YNL256W	-65	4	uORF	uORF4	Yes
WSC3	YOL105C	-50	7	uORF	uORF4	Yes
MKK1	YOR231W	-71	10	uORF	uORF5	No

[a] No uORF with the same position and length in our data set.
[b] Our model predicts uORF3 (in our data set) of gene IMD4 as functional.
[c] uORF identifiers used in the supplementary material for this paper.

8 Discussion

The work presented here uses a machine learning (ML) approach to investigate the regulatory role of uORFs in 5′ UTRs in the yeast *S. cerevisiae*. We are not aware of any previous work of this kind. However, there is other work where machine learning methods have been used to investigate other aspects of post-transcriptional regulation. There is also work using other methods to investigate the regulatory role of uORFs in mammalian species, and work using other computational approaches to investigate the regulatory role of other UTR features in yeast.

Machine learning methods have been used for predicting translation initiation sites. Zeng *et al.* [27] and Tzanis and Vlahavas [22] used feature generation and feature selection with standard ML classifiers such as decision trees, artificial neural networks, naïve Bayes, and support vector machines, while Li and Jiang [9] have used edit kernels for support vector machines. However,

we are not aware of any previous work applying machine learning to the problem of identifying functional uORFs.

Crowe *et al.* [1] have identified uORFs of over 20 codons in length that are conserved in human and mouse genomes. Those uORFs that are conserved between human and mouse are predicted to code for bioactive peptides. They cite studies that suggest that some of these peptides play a role in regulation. In our work we do not place a lower limit on the length of uORFs that are considered, and the prediction model does not depend on sequence conservation across species.

Kwon *et al.* [7] have carried out experimental work to investigate the regulatory role of uORFs and secondary structures in 5′ UTRs. They carried out site-directed mutagenesis studies of human ADH5/FDH and Myf6 genes, measuring the RNA transcripts, investigating the interactions between mRNA and proteins involved in translation, and analysing the RNA secondary structures of the 5′ UTRs. Their results suggest that uORFs and stem-loops in the 5′ UTR can reduce translation of the main coding sequence.

While the related work mentioned above has examined the regulatory role of 5′ UTRs in mammalian species, Ringnér and Krogh [20] have carried out computational studies to investigate the regulatory role of secondary structure in yeast 5′ UTRs. They have computed the folding free energies of the 50 nucleotides immediately upstream of the coding sequence for all verified genes in *S. cerevisiae* and have found that "weakly folded 5′ UTRs have higher translation rates, higher abundances of the corresponding proteins, longer half-lives, higher numbers of transcripts, and are upregulated after heat shock" [20]. One way to extend our study would be to consider additionally the locations of uORFs with respect to predicted secondary structure in the 5′ UTRs.

9 Conclusions and Future Work

In this study, we combined a machine learning technique, ILP, with a bioinformatics tool, ORF Finder, to learn about the upstream Open Reading Frames (uORFs) of *S. cerevisiae*. The ILP approach used in this work provides a way to integrate information derived from genome data with biological knowledge.

We have shown that the positive-only setting of ILP system, CProgol 4.4, can be used to automatically generate rules which identify functional uORFs. The rules are simple and easy to understand. Yet, when the model is used as a predictor, it can make the search for novel functional uORFs 17 times more efficient than using random sampling.

In the future, we would like to investigate whether making background knowledge of RNA structural features available to the ILP learner leads to a better model. Functional uORFs that have been verified so far are conserved in many species [6]. Thus it is worth investigating whether ILP rules can be combined with information on biological conservation, particularly with other yeast species, to refine the model and to test its validity.

10 Acknowledgements

We would like to thank Per Sunnerhagen (Department of Cell and Molecular Biology, Lundberg Laboratory, Göteborg University, Göteborg, Sweden) and Alex Wilson (Division of Mathematics and Statistics, School of Computing, The Robert Gordon University, Aberdeen, UK) for useful discussions.

References

[1] Mark L. Crowe, Xue-Qing Wang, and Joseph A. Rothnagel. Evidence for conservation and selection of upstream open reading frames suggests probable encoding of bioactive peptides. *BMC Genomics*, 7(16), 2006.

[2] Marija Cvijovic. Comparative Genomic Study of upstream Open Reading Frames. [Online] Masters Thesis, Chalmers University of Technology, Available from: http://www.math.chalmers.se/Stat/Bioinfo/Master/Theses/2005/2.pdf, 2005. [Accessed 3 May 2006].

[3] Tania Fiaschi, Riccardo Marzocchini, Giovanni Raugei, Daniele Veggi, Paola Chiarugi, and Giampietro Ramponi. The 5′-untranslated region of the human muscle acylphosphatase mRNA has an inhibitory effect on protein expression. *FEBS Letters*, 417(1):130–134, November 1997.

[4] Chris M. Grant, Paul F. Miller, and Alan G. Hinnebusch. Sequences 5′ of the first upstream open reading frame in *GCN4* mRNA are required for efficient translation reinitiation. *Nucleid Acids Research*, 23(19):3980–3988, 1995.

[5] Alan G. Hinnebusch. Translational Regulation of Yeast *GCN4*. A Window on Factors that Control Initiator-tRNA Binding to the Ribosome. *J. Biol. Chem.*, 272(35):21661–21664, 1997.

[6] Michele Iacono, Flavio Mignone, and Graziano Pesole. uAUG and uORFs in human and rodent 5′ untranslated mRNAs. *Gene*, 349:97–105, 2005.

[7] Hye-Sook Kwon, Dong-Kee Lee, Jae-Jung Lee, Howard J. Edenberg, Yong ho Ahn, and Man-Wook Hur. Posttranscriptional Regulation of Human *ADH5/FDH* and *Myf6* Gene Expression by Upstream AUG Codons. *Archives of Biochemistry and Biophysics*, 386(2):163–171, 2001.

[8] Nada Lavrač and Sašo Džeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, 1994.

[9] Haifeng Li and Tao Jiang. A Class of Edit Kernels for SVMs to Predict Translation Initiation Sites in Eukaryotic mRNAs. *Journal of Computational Biology*, 12(6):702–718, July 2005.

[10] Flavio Mignone, Carmela Gissi, Sabino Liuni, and Graziano Pesole. Untranslated regions of mRNAs. *Genome Biology*, 3(3):reviews0004.1–reviews0004.10, 2002.

[11] David R. Morris and Adam P. Geballe. Upstream Open Reading Frames as Regulators of mRNA Translation. *Molecular and Cellular Biology*, 20(23):8635–8642, 2000.

[12] S. H. Muggleton, C. H. Bryant, A. Srinivasan, A. Whittaker, S. Topp, and C. Rawlings. Are Grammatical Representations Useful for Learning from Biological Sequence Data?-A Case Study. *Journal of Computational Biology*, 8(5):493–521, 2001.

[13] Stephen Muggleton and John Firth. CProgol4.4: a tutorial introduction. In Sašo Džeroski and Nada Lavrač, editors, *Relational Data Mining*, chapter 7, pages 160–188. Springer-Verlag, 2001.

[14] Stephen Muggleton and Luc De Raedt. Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming*, 19(20):629–679, 1994.

[15] Stephen H. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[16] Stephen H. Muggleton. Learning from Positive Data. In Stephen H. Muggleton, editor, *Proceedings of the 6th International Workshop on Inductive Logic Programming*, volume 1314 of *Lecture Notes in Artificial Intelligence*, pages 358–376. Springer-Verlag, 1996.

[17] Graziano Pesole, Flavio Mignone, Carmela Gissi, Giorgio Grillo, Flavio Licciulli, and Sabino Liuni. Structural and functional features of eukaryotic mRNA untranslated regions. *Gene*, 276:78–81, 2001.

[18] Anthony A. Philippakis, Fangxue Sherry He, and Martha L. Bulyk. Modulefinder: A tool for computational discovery of cis regulatory modules. In Russ B. Altman, Tiffany A. Jung, Teri E. Klein, A. Keith Dunker, and Lawrence Hunter, editors, *Pacific Symposium on Biocomputing*. World Scientific, 2005.

[19] Becky M. Pickering and Anne E. Willis. The implications of structured 5′ untranslated regions on translation and disease. *Seminars in Cell & Developmental Biology*, 16(1):39–47, February 2005.

[20] Markus Ringnér and Morten Krogh. Folding Free Energies of 5′-UTRs Impact Post-Transcriptional Regulation on a Genomic Scale in Yeast. *PLoS Comput Biol*, 1(7):e72, 2005.

[21] P. Stothard. The sequence manipulation suite: JavaScript programs for analyzing and formatting protein and DNA sequences. *BioTechniques*, 28:1102–1104, 2000.

[22] George Tzanis and Ioannis Vlahavas. Prediction of Translation Initiation Sites Using Classifier Selection. In G. Antoniou, G. Potamias, D. Plexousakis, and C. Spyropoulos, editors, *Proc. 4th Hellenic Conference on Artificial Intelligence*, 2006.

[23] Cristina Vilela, Bodo Linz, Claudina Rodrigues-Pousada, and John E. G. McCarthy. The yeast transcription factor genes *YAP1* and *YAP2* are subject to differential control at the levels of both translation and mRNA stability. *Nucleid Acids Research*, 26(5):1150–1159, 1998.

[24] Cristina Vilela and John E. G. McCarthy. Regulation of fungal gene expression via short open reading frames in the mRNA 5′ untranslated region. *Molecular Microbiology*, 49(4):859–867, August 2003.

[25] Cristina Vilela, Carmen Velasco Ramirez, Bodo Linz, Claudina Rodrigues-Pousada, and John E. G. McCarthy. Post-termination ribosome interactions with the 5′ UTR modulate yeast mRNA stability. *The EMBO Journal*, 18(11):3139–3152, 1999.

[26] Gavin S. Wilkie, Kirsten S. Dickson, and Nicola K. Gray. Regulation of mRNA translation by 5′- and 3′-UTR-binding factors. *Trends in Biochemical Sciences*, 28(4):182–188, April 2003.

[27] Fanfan Zeng, Roland H. C. Yap, and Limsoon Wong. Using Feature Generation and Feature Selection for Accurate Prediction of Translation Initiation Sites. *Genome Informatics*, 13:192–200, 2002.

[28] Zhihong Zhang and Fred S. Dietrich. Identification and characterization of upstream open reading frames (uORF) in the 5′ untranslated regions (UTR) of genes in *Saccharomyces cerevisiae*. *Current Genetics*, 13(294):1–11, October 2005.