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

<table>
<thead>
<tr>
<th>Title</th>
<th>A first step towards learning which uORFs regulate gene expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors</td>
<td>Bryant, CH, Kemp, GJL and Cvijovic, M</td>
</tr>
<tr>
<td>Type</td>
<td>Article</td>
</tr>
<tr>
<td>URL</td>
<td>This version is available at: <a href="http://usir.salford.ac.uk/1755/">http://usir.salford.ac.uk/1755/</a></td>
</tr>
<tr>
<td>Published Date</td>
<td>2006</td>
</tr>
</tbody>
</table>

USIR is a digital collection of the research output of the University of Salford. Where copyright permits, full text material held in the repository is made freely available online and can be read, downloaded and copied for non-commercial private study or research purposes. Please check the manuscript for any further copyright restrictions.

For more information, including our policy and submission procedure, please contact the Repository Team at: usir@salford.ac.uk.
A First Step towards Learning which uORFs Regulate Gene Expression

Selpi¹, Christopher H. Bryant¹, Graham J.L. Kemp², Marija Cvijovic³

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

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

³Max 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].
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-
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 \textit{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 (\(\approx 4\) 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 \(\approx 20\) 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
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 1: Detailed composition of prediction made by ORF Finder [21]**

<table>
<thead>
<tr>
<th>Number of Genes</th>
<th>Transcribed uORFs</th>
<th>Other uORFs (Not known if transcribed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Functional</td>
<td>Non-functional</td>
</tr>
<tr>
<td>17 studied genes</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>5,585 other genes</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5,602 genes</td>
<td>20</td>
<td>2</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Gene Name</th>
<th>Systematic Name</th>
<th>Transcribed uORFs</th>
<th>Other uORFs</th>
<th>Positive Examples</th>
<th>Negative Examples</th>
<th>Random Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLN3</td>
<td>YAL040C</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>GCN4</td>
<td>YEL009C</td>
<td>4</td>
<td>15</td>
<td>4</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>HAP4</td>
<td>YKL109W</td>
<td>2</td>
<td>26</td>
<td>2</td>
<td>-</td>
<td>26</td>
</tr>
<tr>
<td>TIF4631</td>
<td>YGR162W</td>
<td>5</td>
<td>202</td>
<td>5</td>
<td>-</td>
<td>202</td>
</tr>
<tr>
<td>YAP1</td>
<td>YML007W</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>YAP2</td>
<td>YDR423C</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HOL1</td>
<td>YNR055C</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>PET111</td>
<td>YMR257C</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>SCO1</td>
<td>YBR037C</td>
<td>1</td>
<td>4</td>
<td>-</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>CBS1</td>
<td>YDL069C</td>
<td>1</td>
<td>4</td>
<td>-</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>INO2</td>
<td>YDR123C</td>
<td>1</td>
<td>9</td>
<td>-</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>PPR1</td>
<td>YLR014C</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>URA1</td>
<td>YKL216W</td>
<td>1</td>
<td>13</td>
<td>-</td>
<td>-</td>
<td>14</td>
</tr>
<tr>
<td>LEU4</td>
<td>YNL104C</td>
<td>1</td>
<td>12</td>
<td>-</td>
<td>-</td>
<td>13</td>
</tr>
<tr>
<td>RCK1</td>
<td>YGL158W</td>
<td>2</td>
<td>49</td>
<td>-</td>
<td>-</td>
<td>51</td>
</tr>
<tr>
<td>DCD1</td>
<td>YHR144C</td>
<td>1</td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>18</td>
</tr>
<tr>
<td>SCH9</td>
<td>YHR205W</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
</tbody>
</table>

17 Genes: 30 380 20 2 388

*Names are taken from SGD (http://www.yeastgenome.org).

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.

\[
\text{uORF}(X, Y, Z) \quad \text{where} \quad X \text{ is a uORF ID, } Y \text{ is the distance of } X \text{ from the start of coding sequence, and } Z \text{ is the length of } X.
\]

\[
\text{utr}(X, Y, Z) \quad \text{where} \quad X \text{ is a UTR ID, } Y \text{ is the number of uORF that } X \text{ has, and } Z \text{ is the intergenic sequence length between } X \text{ and the previous gene.}
\]

\[
\text{has_uORF}(X, Y) \quad \text{where} \quad X \text{ is a UTR ID and } Y \text{ is the uORF ID of one of } X\text{'s uORFs.}
\]

\[
\text{belongs_to}(X, Y) \quad \text{where} \quad X \text{ is a uORF ID and } Y \text{ is a UTR ID to which } X \text{ belongs.}
\]

\[
\text{context}(X, Y, Z) \quad \text{where} \quad X \text{ is a uORF ID, } Y \text{ and } Z \text{ are the frequencies of AU and GC within 20 bases downstream of } X\text{'s stop codon.}
\]

\[
\text{up_context}(X, Y, Z) \quad \text{where} \quad X \text{ is a uORF ID, } Y \text{ and } Z \text{ are the frequencies of AU and GC within 20 bases upstream of } X\text{'s start codon.}
\]

Table 4: Intensional Background Knowledge$^a$

\[
\text{has_shortest_dist_in_UTR}(UORF):= \text{uORF(UORF,ShortestDist,_)}, \text{belongs_to(UORF,UTR)}, \text{setof(Dist,(has_uORF(UTR,UORFX),uORF(UORFX,Dist,_)),List)}, \text{List = [ShortestDist|_]}. 
\]

\[
\text{has_shortest_len_in_UTR}(UORF):= \text{uORF(UORF,_,ShortestLen)}, \text{belongs_to(UORF,UTR)}, \text{setof(Len,(has_uORF(UTR,UORFX),uORF(UORFX,_,Len)),List)}, \text{List = [ShortestLen|_]}. 
\]

\[
\text{gcrich_down_up}(UORF):- \text{context(UORF,Au,Gc), Gc > Au, up_context(UORF,A,G), G > A.}
\]

\[
\text{gcrich_down_aurich_up}(UORF):- \text{context(UORF,Au,Gc), Gc > Au, up_context(UORF,A,G), G < A.}
\]

\[
\text{aurich_down_up}(UORF):- \text{context(UORF,Au,Gc), Gc < Au, up_context(UORF,A,G), G < A.}
\]

\[
\text{gcrich_up_aurich_down}(UORF):- \text{context(UORF,Au,Gc), Gc < Au, up_context(UORF,A,G), G > A.}
\]

$^a$CProgo1'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

模体模式（modeh）声明了用于假设头的规则，而模体模式（modeb）声明了用于假设体的规则。

5 Generating a Model that Identifies Functional uORFs

我们研究ILP是否能够自动生成一个模型来识别功能性uORFs，并且这个模型，当作为过滤器使用时，是否能够比随机采样更有效。

训练集包含14个正例和259个随机样本，测试集包含6个正例和129个随机样本。

CProgol的参数设置如下：
- **posonly** 只对正例和随机样本来学习；
- **inflate** 给定数据/谓词一个权重（一般为4,200%）；
- **c** （规则体最大原子数）设置为6；
- **nodes** （规则体的最大深度）设置为7,000；
- **r** （证明时的解析深度的最大值）设置为700。

我们将CProgol 4.4的假设空间定义为可以构造目标谓词 has_functional_role/1 的定义。这是通过给定模式声明（见表5）来完成的。

图3显示了生成的模型。

6 Measuring Model Performance using Relative Advantage

一个独立的测试集用于评估模型。CProgol 4.4的默认性能衡量标准是预测准确率。然而，当正例很少时，这种衡量标准在预测准确性上给出的估计值是不准确的。因此，我们定义了相对优势度量来衡量模型的性能。
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)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Positives correctly classified as positives</td>
<td>3</td>
</tr>
<tr>
<td>Randoms falsely classified as positives</td>
<td>4</td>
</tr>
<tr>
<td>Positives falsely classified as randoms</td>
<td>3</td>
</tr>
<tr>
<td>Randoms correctly classified as randoms</td>
<td>125</td>
</tr>
<tr>
<td>mean RA</td>
<td>17.3</td>
</tr>
</tbody>
</table>

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

\[ RA = \frac{A}{B}; \]  

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-
region 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].

<table>
<thead>
<tr>
<th>Gene Name</th>
<th>Systematic Name</th>
<th>uORF’s Position</th>
<th>uORF’s Length</th>
<th>uORF Identifier</th>
<th>Predicted as Functional</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARV1</td>
<td>YLR242C</td>
<td>-125</td>
<td>12</td>
<td>uORF1 uORF5</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-108</td>
<td>3</td>
<td>uORF2 uORF4</td>
<td>Yes</td>
</tr>
<tr>
<td>ECM7</td>
<td>YLR443W</td>
<td>-40</td>
<td>7</td>
<td>uORF3 uORF6</td>
<td>Yes</td>
</tr>
<tr>
<td>HEM3</td>
<td>YDL205C</td>
<td>-129</td>
<td>9</td>
<td>uORF uORF8</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-60</td>
<td>4</td>
<td>uORF uORF5</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-11</td>
<td>4</td>
<td>uORF uORF7</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-42</td>
<td>5</td>
<td>uORF uORF4</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-70</td>
<td>7</td>
<td>uORF uORF5</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-27</td>
<td>5</td>
<td>uORF uORF3</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-41</td>
<td>6</td>
<td>uORF uORF5</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-25</td>
<td>4</td>
<td>uORF uORF3</td>
<td>Yes</td>
</tr>
<tr>
<td>IMD4</td>
<td>YML056C</td>
<td>-99</td>
<td>14</td>
<td>uORF -</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-110</td>
<td>24</td>
<td>uORF1 uORF11</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-84</td>
<td>6</td>
<td>uORF2 uORF9</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-70</td>
<td>4</td>
<td>uORF3 uORF10</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-65</td>
<td>4</td>
<td>uORF uORF4</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-50</td>
<td>7</td>
<td>uORF uORF4</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-71</td>
<td>10</td>
<td>uORF uORF5</td>
<td>No</td>
</tr>
</tbody>
</table>

*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


