Delft University of Technology

Masters Thesis

Inferring features from 5'UTR sequences to Translation Initiation Rates in S.cerevisiae

Author:
Eftychia Thomaidou
student number: 4182219

Supervisors:
Marcel Reinders
Alexey Gritsenko

in the

Computer Science

August 2017

The source code of this research can be found under the repository
https://bitbucket.org/EftychiaThomaidou/thesis/src
Access can be requested to eftychiathomaidou@gmail.com

Contents

Chapter 1

Introduction

From the *Central Dogma of molecular Biology* [1], we know how the genetic information passes from different levels, that of *DNA* transcribed into *RNA*, that is then translated into *protein*. So, the actual products that a living organism produces, are the proteins and thus the translation, the process of synthesizing those proteins, is of high significance. That is because a better understanding and manipulation of this process has significant affect in various fields. Just imagine the consequences of the fine tuning of gene regulation in personalized medicine. Or the use of it in the field of biotechnology and synthetic biology.

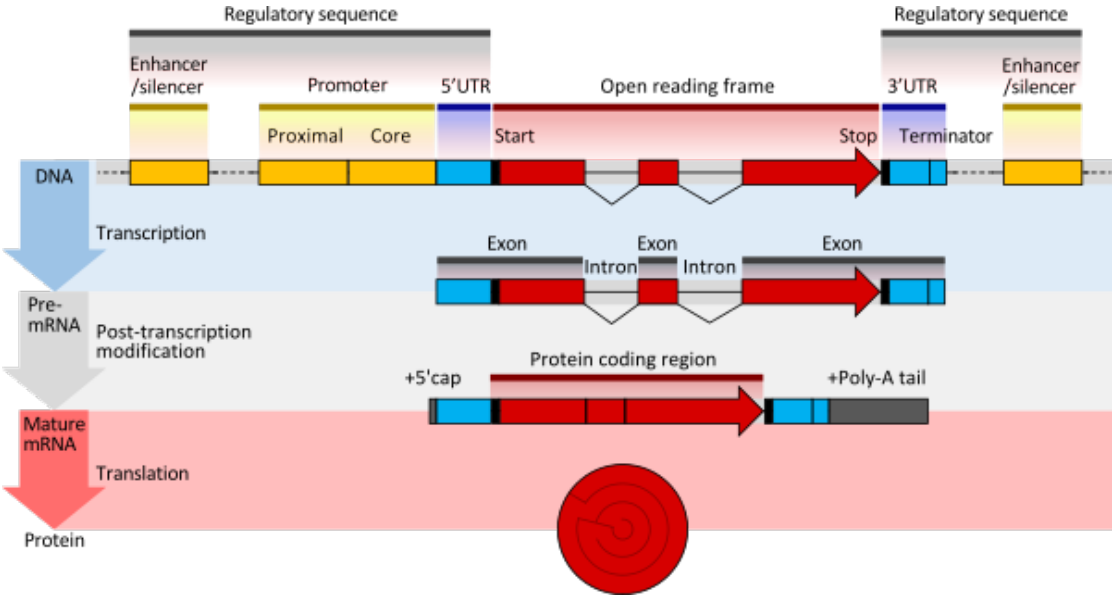


FIGURE 1.1: Central Dogma of molecular Biology. DNA transcribes to RNA, and RNA translates to protein. From [2]

Let us focus at the cytoplasmic mature mRNA, Figure 1.2. As it is known from the literature [3–7], the *5' Untranslated Region*, known as 5'UTR, of mRNA although is not translated itself, may contain elements, that influence the gene expression. As 5'UTR, is named the region just

in front of the *coding sequence*, which is the part of *mRNA* that is translated into *protein*, and towards the 5 *prime end*.

FIGURE 1.2: mRNA structure

However, the exact role of the 5'UTR is unknown or not confirmed. The same counts for its regulatory elements, so it is of high interest to be discovered. In this research, we studied the impact of the 5'UTR sequences on translation. This is done by generating various features describing the 5'UTR. Those features are then used as input in regression and classification models, that have the *Translation Initiation Rates* as target. What we define as *Translation Initiation Rate* is the speed (or rate), at which the protein production, from mRNA, occurs. The reason why it is of such importance, is dual. To begin with, it is very useful to be able to predict the initiation rates for new sequences and consequently be able to synthesize new sequences with high initiation rate [3, 5, 6]. Additionally, it is important to understand which of the elements, located in the 5' UTR, influence the translation initiation rates and thus the translation.

Aim of this research is detecting those features from the 5' UTR of yeast's mRNA, that lead to higher translation initiation rates. In order to achieve this, data mining and machine learning techniques were used to build predictive models. In the following chapters, we describe related research, that has motivated this topic and supplied us with data. As a next step we get a better insight in the data, as well as in the various features that were extracted from the 5'UTR region, and that feed our models. Next we present the feature importance and correlation visualization techniques used in this research. Following, we elaborate about our predictive modeling that includes regression and classification models and as a next step we evaluate the results of our models. Finally, we conclude with the results and the discussion.

Chapter 2

Related Research

This research is motivated by the work of Gritsenko et al. [8]. In this paper, is described the prediction of *translation initiation rates* with the use of *Ribosome Profiling*. RP is a sequencing-based technique [9], that allows us to have snapshots of the locations and the activity of the ribosomes, while they perform the translation. More specifically, they describe a way to use the RP data, that has been used for the per-codon translation elongation and per-gene translation initiation rates. For that, they use the *TASEP*, *Totally Asymmetric Exclusion Process*, a simple dynamic model of translation. This method was introduced in 1970, by Frank Spitzer, in Interaction of Markov Processes [10].

In this model, the translation of mRNA by the ribosome is modeled as a one-dimensional process, in which the ribosome is attracted to the mRNA (initiation rate), and every codon specifies how effective the translation step is (elongation rate). The process of the translation following the TASEP manner, can be depicted in the Figure 2.1. The aforementioned process can be achieved by the division of every coding sequence per gene into segments. Ribo-seq reads, $R_{[l,r]}$, and mRNA-seq reads, $M_{[l,r]}$ are mapped to these segments.

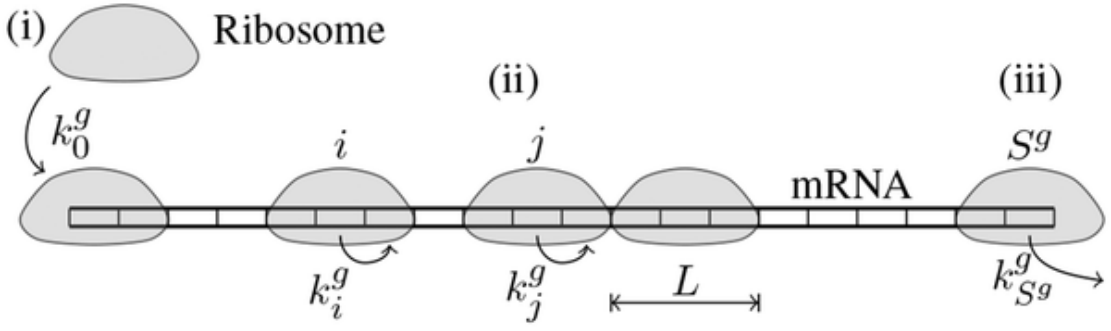


FIGURE 2.1: In the above image, is shown the TASEP methodology as applied by Gritsenko et al. [8]. In TASEP, mRNAs are modeled as one-dimensional lattices of S^g sites, codons, and ribosomes, as particles occupying L sites (where L = 3 in the figure).

As a next step, they calculate the *ribosome density* $\omega_[l,r]$, where l and r are the starting and ending positioning of the segment, respectively. Where the definition of $\omega_[l,r]$, is:

$$\omega_[l,r] = \frac{d^{Ribo}_{[l,r]}}{d^{mRNA}_{[l,r]}} = \frac{\frac{R_{[l,r]}}{L_{[l,r]}N_R}}{\frac{M_{[l,r]}}{L_{[l,r]}N_M}} \tag{2.1}$$

$R_{[l,r]}$, describes the number of ribo-seqs in a segment, normalized by the total number of ribo reads aligned to all coding sequences. Similarly, $M_{[l,r]}$ represents the mRNA-seq reads.

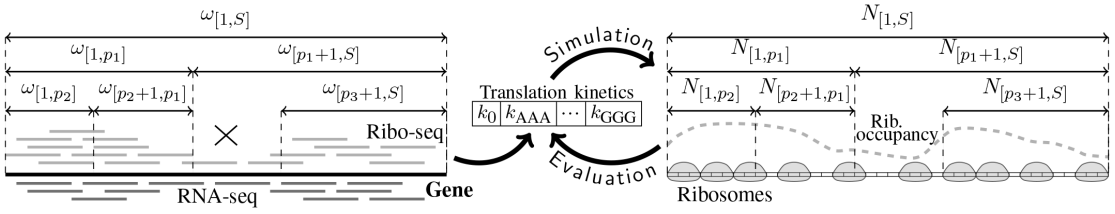


FIGURE 2.2: Schematic overview of the approach inferring translation kinetics from RP data, as it is proposed by Gritsenko et al. [8]

The TASEP model is fitted to these ribosome density profiles, with the use of a *Genetic Algorithm*, GA. A more elaborate scheme is shown in Figure 2.2.

Our research has also been motivated by the work of Ciandrini et al. [11]. Here again, there is use of genome-wide experimental data of ribosomal density on mRNAs for *Saccharomyces Cerevisiae* to translation initiation rates. That is succeeded with the use of a stochastic model, that describes the ribosome traffic dynamics during translation elongation. According to Ciandrini et al., the presence of secondary structures, on the mRNA sequences, inhibits the ribosome to easily bind on the 5' leader composition. That prompted us to the derivation of the *Minimum Free Energy* as a feature in our research. In the work of Ciandrini et al., is also stated that the codon arrangement seems to play a significant role in the determination of the translational efficiency. Following the same assumption, we wonder whether there is a correlation between our target, which is the *translation initiation rates*, and a sequential pattern within the 5'UTR. Lastly, they have observed a correlation between the length of the *Open Reading Frame*, ORF, and the translation initiation rates. Thus, we have included the length into our features as well.

Chapter 3

Experimental Setup

The aim of this research is to provide a better insight, into the relations, and possible causalities, between the *5 prime untranslated regions* and the *translation initiation rates* of those genes. In other words, we are interested to see whether there is any information in the 5' UTR sequences and whether this information correlates to the rate that the sequence is translated. Using the *Extract, Transform and Load*, ETL, paradigm, and a wide variation of data mining and machine learning techniques the relations within the data are explored.

In Section 3.1 the data sets provided and additional data sets used, are explained into detail. In the next section, Section 3.2, can be found the feature extraction from the DNA sequences. Those features are explored in Section 3.3, with the use of exploratory data mining. Further data mining experiments, are explained in Section 3.4, where the prediction of the target, the Translation Initiation Rates, is discussed. The evaluation of our methods is discussed in Section 3.5.

3.1 Data

For the completion of this research, we have gathered data from multiple resources.

We have based our research on the genome of *Saccharomyces Cerevisiae* and that is because yeast, although a simple eykariotic organism, has many essential cellular processes very similar to the human genome [12, 13]. Therefore, by studying the yeast genome we are a step closer to understanding the human genome. Additionally, yeast was the first eykariotic organism to have its genome sequenced [14]. This leads us to a plethora of data and research material to access and double validate our research.

The names of the genes and their initiation rates are based on two different data sets. The first one, is that of Gritsenko et al. [8] and the second one is the one of Ciandrini et al. [11]. Since our research is focused on the 5 prime untranslated region, that is located before the actual coding sequence, we retrieved those regions as well. This data is coming from Nagalakshmi et al. [15] and Yassour et al. [16]. In which the starting, as well as the ending position, of the 5'UTR are

defined. The 5'UTR sequence was subsequently derived from the Yeast Genome database [17]. In our experimental setup the initiation rates gathered are used as the target of our models. It should be noted that these initiation rates are coming from stochastic models, which include noise and imprecisions and thus make our prediction task harder.

For the calculation of the Minimum Free Energy, there was use of the RNAfold tool, available by the ViennaRNA kit [18]. Additionally, Gene Annotation, for the reference to the*response to stress* annotation, was derived from the *Gene Ontology* [19, 20] database.

Following, additional data were added in the Gritsenko et al. dataset [8]. This includes the *average mRNA reads*, the *average Ribosome reads*, the *Fitness* and the *number of Segments per gene*. Those values were used by Gritsenko et al. in order to derive the translation initiation rates in their research [8]. In our research we include them as features in order to verify the propriety of our experimental approach.

Our data contains both features and target values, the translation initiation rates. In order to perform data mining and predictive modeling activities, preprocessing and features extraction are necessary. While the data contain mostly DNA sequences, most data mining and machine learning algorithms work with numerical and categorical data.

3.2 Features Extraction

Aim of this research is to predict translation initiation rates from the 5'UTR sequences. However, we cannot just simply feed our predictor with these sequences. That is because the 5'UTR sequences are not of similar length. Furthermore, 5'UTR sequences are not aligned to each other. Thus, as a first step, we extract generic features from those sequences and those features are then used to train our models. Following, are described the features that were used in our work.

Length : The first feature, that was taken into account, is the length of the sequence. We thought of the length, because according to the literature [5, 11], genes with short coding sequence tend to get translated more often than the longer ones. Maybe such a negative correlation could be seen for the length of the 5'UTR as well. We have concluded to the same assumption about length, during the calculation of the *Conditional Entropy* as well. See Figure 4.10. For the calculation of the length are used the last 100bp upstream and the first 40bp of the coding sequence. In other words, the length of a sequence can be of maximum 140 base pairs. We do not use the whole size of the 5'UTR, as it might be expected, that is due to our attempt to normalize our data. Additionally, the *Ribosomal Binding Site*, RBS, is likely to be located approximately 8bp before the *starting codon* according to N. Malys [21].

When a *ribosome* docks on the *mRNA* to begin the translation, it searches for a particular pattern on the upstream sequence, where it can land [6]. This particular piece of sequence is called *RBS*. While we are searching for this pattern in the upstream sequences, as well for patterns that

enhance the docking of the ribosome on the mRNA, we came up with the following features, that are based on the actual content of each sequence. Those features are: the *Frequency of Base Pairs*, the *Frequency of 2-mers* and the *Frequency of 3-mers*.

Base Pair Frequency : For the calculation of these features, is used the aggregation of As found in each sequence, divided by the actual length of that sequence. In a similar manner is calculated the frequency of Ts, Gs and Cs.

$$\frac{\sum b}{length}, b \in A, T, G, C \tag{3.1}$$

2-mers Frequency : These are 16 features, that represent the frequency of occurrence of every possible combination using two base pairs; for example AA, AT, AG and so on. For the calculation is used the number of counts divided by length-1.

$$\frac{\sum b}{length - 1}, b \in AA, AT, AG, AC, ..., CC \tag{3.2}$$

3-mers Frequency : In a similar manner, those 64 features represent combinations of three base pairs; like AAA, AAT, ATA and so on. The features show the frequencies of the 3-mers, which is the number of counts divided by length-2.

$$\frac{\sum b}{length - 2}, b \in AAA, ATA, .., CCC \tag{3.3}$$

From those 84 features we would like to see a correlation between a pattern of bases and the translation initiation rates.

While trying to infer to features related to translation rates, we are seeking our answers in what is known in biology. Where we find, that translation responses seem to be strongly influenced by the gene function. Subsequently, we focus our interest in the genes with the biological process *response to stress*. As it is known, genes that are involved in the Response to Stress, get translated more often than others, as they get higher priority [22–24]. Thus, we were motivated to use *stress* as a feature in our algorithm. We would like to see a strong correlation among genes, that are related to *stress* and the *translation initiation rates*.

Stress : According to *Gene Ontology* [19, 20], Response to Stress is "any process that results in a change in state or activity of a cell or an organism (in terms of movement, secretion, enzyme production, gene expression, etc.) as a result of a disturbance in organismal or cellular homeostasis, usually, but not necessarily, exogenous (e.g. temperature, humidity, ionizing radiation)". In order to proceed to the generation of this feature, we used the *Gene Ontology Annotation*. With the use of the *AmiGo* on-line database [19, 20], we were able to find all the descendants of the term *Response to Stress*, (GO:0006950). We used the aggregation of terms associated to each gene to create our feature. In other words, the value of a gene for this feature is a positive natural number, that indicates how many terms related to Stress, a gene has.

Due to secondary structures that are created in the 5' region, is harder for the ribosome to bind on the RBS, and this leads to low translation initiation rates. For that reason we considered the *Minimum Free Energy*, MFE, as a feature. The lower the *free energy* is, the more tight and stable the secondary structure of the mRNA is. In others words, in order to achieve frequent gene translations, we need weak structures [4, 11]. What we aim with this feature, is to see a positive correlation between genes weakly folded and the translation initiation rates.

MFE : For the calculation of the MFE, is used *RNAfold*, available by the ViennaRNA kit [18]. In order to proceed to the actual calculation, is used the whole length of the sequence as it is used for the calculation of the length. In other words, for each sequence are used the last 100bp upstream and the first 40bp of the coding sequence.

3.3 Exploratory Data Mining

In order to get a better idea about how helpful our features are, as well as how well they associate individually or together against the translation initiation rates, we followed two strategies. These strategies are: a) inspection of the correlation between the feature values and the initiation rates, and b) data visualization by reduction of the dimensionality.

Feature Importance and Correlations

While aiming to observe how informative the individual features are, we considered their correlation with the translation initiation rates. For the feature correlation calculation, we have used the *Pearson product-moment correlation coefficient*, also known as *Pearson's r* [25], and the *Spearman's rank correlation coefficient*, known as *Spearman's* ρ [26]. The Pearson's r method describes the linear relationship between two true values. Whereas the Spearman's ρ method, uses ranks and evaluates how well the relationship of two variables can be described by a monotonic function. Because of the different approach the two methods use, for the calculation of the correlation, it has been decided that both methods should be used in this research, as their combination leads to better understanding of the data. Feature importances are calculated by looking at the ordering of the features, that each regressor tree of the Random Forest uses, in order to split the target efficiently. The more a feature is used for this division, at the beginning of a regression tree, the more influence this feature has on the final outcome and thus this feature is of higher importance.

For the calculation of the feature importance we have used the attribute *feature_importances_* given by the library *sklearn.ensemble.RandomForestRegressor*[27]. According to which, is evaluated the importance of a variable X_m for predicting Y. This is achieved by adding up the weighted impurity, decreases $p(t)\Delta i(s_t, t)$ for all nodes t, where X_m is used, averaged over all N_T trees in the forest:

$$Imp(X_m) = \frac{1}{N_T} \sum_{T} \sum_{t \in T : v(s_t) = X_m} p(t) \Delta i(s_t, t) \tag{3.4}$$

Where $p(t)$ is the proportion of N_t/n of samples reaching t. $v(s_t)$ is the variable used in the split s_t. This measure is also known as *Gini importance* or *Mean Decrease Impurity*, (MDI) [28, 29].

Visualization Techniques

To gain insight into the datasets, visualization techniques for high dimensional data sets were used. With the use of *Multidimensional Scaling*, MDS [30], the multidimensional data can be scaled to a 2D projection and then visualized as plot using a distance measure, such as the *Manhattan distance*. In order to feed the MDS, we have first calculated the *Distance Matrix* for each dataset. The Distance Matrix has been calculated with the use of the Manhattan distance, which is simply the summation of the absolute pairwise difference.

$$d(a, b) = \sum_{f=1}^{m} |a_f - b_f| \tag{3.5}$$

Where a and b are two different samples, so two sequences among n sequences in our data. f represents the feature in a dataset of m features. The end result is a similarity matrix $n \cdot n$.

The resulting 2D plot, that can be seen in Figure 4.4, tells us how similar our samples are. In other words, we can see how close they lie together in this 2D space. This can give us an insight into the separability of the data. The target of our dataset, the translation initiation rates, can be used to color the final scatter plot to visualize the distribution of the initiation rates among the different records.

Other algorithms to visualize high dimensional data sets are *Andrews' Curves* (Figures: 4.6a and 4.6b), *RadViz* plots (Figures: 4.7a and 4.7b) and *Lag* plots (Figures: 4.5a and 4.5b). *Andrews' Curves*, are similar to *Parallel Coordinates* plots [31], but they differ in that they show more smooth behavior and therefore are usually easier to interpreet. In Andrews' Curves, each sequence of length l in the dataset is transformed into a polynomial of degree n, with the values of the record as coefficients.

3.4 Predictive Modeling

Predicting or estimating, the exact values of the Translation Initiation Rates, from the features gathered, as described in Section 3.2, is the primary goal of this research. However, the prediction of the translation initiation rates given the 5'UTR sequences is not a trivial task. Especially, when the given data is output of a stochastic model. The prediction of the translation initiation rates, can be seen as a Regression problem, where we predict the exact initiation rate, given the features per record. On the other hand, in case we want to know whether the translation

initiation rate of a sequence is relatively high or low, the task can be seen as a Classification problem.

In this section, we elaborate on the predictive models used in our various experiments and the way we have measured the quality of these experiments. For higher reliability in our results, we have used a 5 fold *Cross Validation* with an additional separate validation set. For the preprocessing of the feature values, has been used a *Standard Scaler*, before applying the regression model. The Standard Scaler standardizes every feature separately, by removing the mean and scaling to unit variance.

3.4.1 Regression Methods

A wide variety of regression methods and their parameters were utilized, in order to see how well the target could be predicted. The regression algorithms used are explained below.

Random Forest Regressor

One of the models that has been used, is the *Random Forest Regressor*. An outline of this algorithm is given in Algorithm 1. Briefly, this method picks up, in every iteration, a random selection of features and creates *decision trees* as many as the variable *n_estimators* is set. In a regression decision tree, we fit a regression model to the target variable, by using each of the independent variables. Then for each independent variable, the data is split at several split points. At each split point, the *Mean Squared Error*, MSE, is calculated, between the predicted value and the actual value [27, 32]. The definition of the MSE can be found in Section 3.5.2. The variable resulting in minimum MSE is selected for the node. Then this process is recursively continued, till either a) all leaves contain one value, or b) one of the stopping criteria is met, such as maximum depth. In the end, we end up with n variables, where n equals the n_estimator predicted values. In order to arrive at a final estimate, the regressor takes the average value and calculates *the coefficient of determination* against the test and the validation set. The definition of the coefficient of determination can be found in Section 3.5.2.

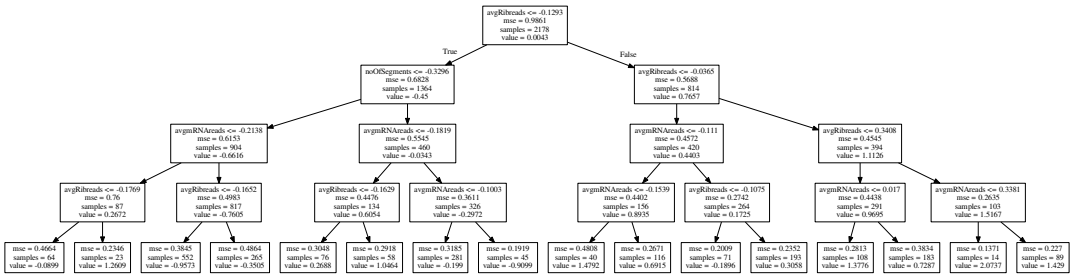


FIGURE 3.1: Decision Regression Tree computed for Gritsenko et al. dataset [8]. max_depth=4

Algorithm 1 Random Forest for Regression as defined by T. Hastie et al. [33]

1. For b=1 to B:

   (a) Draw a bootstrap sample Z^* of size N from the training data.

   (b) Grow a random forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum size n_{min} is reached.

      i. Select m' variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Spit the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}^B{}_1$.

To make a prediction at a new point x:
$Regression: f(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$

Decision Tree Regressor

The *Decision Tree* regressor is a very similar method to the random forest regressor, only that in this case we have only one tree, instead of a forest. In a decision tree, at every node we split the data into smaller subsets. Please refer to Figure 3.1. To begin with, for every feature we find all possible split points (thresholds). For each threshold and feature combination, is calculated the *MSE* against the target value and is selected the split with the minimum error. Then the selected feature and threshold combination is used to split the dataset, and the feature and threshold combination is saved in the node. Now depending on whether the target values are bigger or smaller than the threshold, the data either belongs to the right or left child respectively. This is an iterative process, that stops when the algorithm meets one of the following criteria. Those criteria are a) the max_depth, b) the min_samples_leaf and c) the node consists of only one value. Predicting an unseen record is done by following a path through this tree, until we arrive at a leaf node. The prediction is then the mean of the target values assigned to this leaf node.

SVM

Support Vector Machine Regression [34], is a regression method, that tries to fit a hyperplane on the data. In order to achieve this, the SVM places the hyperplane among the data in such a way that the margin between the plane and the data is optimized. SVM is a linear regressor. By using kernels, the method can be applied on non-linear data as well. The kernel can be set as a parameter and is found that, in our case, the *Radial basis function kernel*, RBF kernel, performs best. The RBF kernel fits simple models in a region local to the target point x_0. The RBF kernel is defined by the following formula:

$$K(a,b) = \exp\left(-\frac{\parallel a-b \parallel^2}{2\sigma^2}\right) \tag{3.6}$$

Where a and b are gene samples in our data. In our experiments we used the implementation given by scikit-learn [35] with the RBF kernel and empirically optimized parameters.

Kriging

Gaussian process regression also known as *Kriging* [36], is a popular regression method often used in geostatistics. Under the right assumptions, Kriging gives the Best Unbiased Linear Predictor. It computes not only the predicted mean, but also provides a prediction error also known as the Kriging variance. A major downside of Kriging, is that the execution time is $O(n^3)$ in the number of records and the memory complexity is $O(n^2)$. Kriging also uses a kernel like Support Vector Machine. The standard kernel used is the Gaussian squared exponential kernel and is defined as:

$$K_{SE}(a,b) = \sigma^2 \exp\left(-\gamma d(\frac{a}{l},\frac{b}{l})\right)^2 \tag{3.7}$$

Where l is a length-scale parameter either of length one or the number of dimensions m, of the input dataset. This parameter can either be set or is estimated by means of the *maximum likelihood* procedure of the Kriging model. The implementation used in our experiments comes from the scikit learn package version 0.17 [35], using the default parameters.

LASSO

LASSO is the abbreviation for the *Least Absolute Shrinkage and Selection Operator*, which is a regression method, that aims in shrinking the feature coefficients, while reducing the model complexity. The latter occurs because LASSO picks randomly a feature among those with the highest correlation and reduces the coefficients of the rest to zero. The features with coefficient equal to zero are excluded, performing thus a feature selection. LASSO minimizes the function as given in Equation 3.8, where α can be tuned by the user. Setting α to one, is equivalent to an ordinary least square.

$$\min_w \frac{1}{2n} \parallel Xw - y \parallel_2^2 + \alpha \parallel w \parallel_1 \tag{3.8}$$

Where y is the real translation initiation values, that we try to predict. $\| w \|_1$ is the $l1$-norm (Manhattan distance) of the parameter vector, w.

3.4.2 Classification Methods

Since, predicting the exact value of the Initiation Rates seems to be a difficult task, we also chose to perform *Classification*, in order to see whether the data are separable in two classes. Those classes are the Low or High translation initiation rates. For the training data set, we obtained these labels by dichotomizing the translation initiation rates. All translation initiation rate values above the mean value, across all genes, are considered as High; whereas those with values below the mean are considered to be Low. Thus, we have two classes, $label - 1$ and $label - 0$, respectively.

In order to perform the classification, there were two different kinds of experiments performed. In the first, we split the dataset into two classes using the mean initiation rate as boundary. And in the second, only the extreme values for the translation initiation rates were taken into account.

Using the auto-sklearn class [37], many classifiers and machine learning pipelines can be run, in order to optimize the AUC score. Please refer to the Quality Metrics section about the AUC, Section 3.5.2. The final predictor and ROC curves are shown and discussed in Section 4.2.

For the classification task, mainly a *Gradient Boosting Classifier* and a *Support Vector Classifier*, SVC, were used. The parameters of the Gradient Boosting Classifier are learned by using the Auto-sklearn optimizer. The parameters used for the Gradient Boosting Classifier are: n_estimators=392, learning_rate=0.062, max_depth=3, random_state=0, max_features=3, loss='deviance', min_samples_leaf=7, min_samples_split=7.

The SVC was used with the default parameters, with the aim to see whether a Support Vector Machine could obtain better results. The default parameters are the following: C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, verbose=False, max_iter=-1.

Gradient Boosting Classifier

The *Gradient Boosting Classifier* is a prediction method, that forms an ensemble of decision trees using a forward stage-wise way. At each stage, a base model, which is an ensemble of classification trees, is fit to the residual of the current model. Where the residual is the gradient of the binomial deviance loss function. In other words, in every iteration the target is the binomial deviance of the target (0 - 1) and the prediction output of the current model. The binomial deviance is defined as:

$$yP - log(1 + e^P) \tag{3.9}$$

Where P is log odds, $log\left(\frac{P(c)}{P(\neg c)}\right)$, of a sample belonging to a class c.

Support Vector Classifier

The *Support Vector Classifier*, SVC, is searching for the hyperplane, that creates the biggest margin between training points for the classes 0 and 1. SVC is using only a subset of the training data and that is because the cost function that is used for this model, does not care about training points that lie beyond the margin. The optimization problem capturing this concept is the following: $max_{\beta,\beta_0,\|\beta\|=1} M$ is subject to

$$y_i(x_i^T\beta + \beta_0) \geq M, i = 1, ..n \tag{3.10}$$

Where $x^T\beta + \beta_0$ is the hyperplane, x_i^T is a transposed sequence (data point), β is the unit vector and when $\beta_0 = 0$ this hyperplane passes from the origin. $\| \beta \| = 1$ and the maximal margin of width is $2M = 2/\| \beta \|$.

3.5 Evaluation

For the evaluation of the models, used in this research, we used a train-test procedure and quality metrics, that are explained below.

3.5.1 Train-Test Procedure

In order to have more reliable results and avoid the overfitting of the regression models we have trained the models using a five-fold cross validation. Where one fold is used as the test data, which is used to tune the model parameters. The reason why we chose for cross validation is because re-sampling leads to better estimation of the accuracy of the model.

Before applying the cross validation, we have separated a 10% of the complete data in order to form the validation set. The validation set is used as a last step, in order to evaluate the learned models on the validation dataset and get a final objective idea of how well the models perform on unseen data.

3.5.2 Quality Metrics

For the validation of the trained predictive models, several quality indicators were used.

Coefficient of Determination : The *coefficient of determination* is a very common performance measurement. Is denoted as R^2 score and is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - p_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.11}$$

Where y_i denotes the target value and p_i the predicted value. $\bar{y}$ is defined as the mean of the target values. Subsequently, the best value for the R^2 error, would be equal to one.

Mean Absolute Error : The *Mean Absolute Error*, MAE, shows the error made by the predictor in estimating the target. Thus, the lower the MAE is, the better. Note that this measurement is depending on the range of the target data.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - y_i| \tag{3.12}$$

Where the mean absolute error is the average of the absolute errors $|p_i - y_i|$. While, p_i is the prediction and y_i is the true value.

Mean Squared Error : The MSE computes the mean square error, a risk metric corresponding to the expected value of the absolute error loss or $l1$-norm loss.

$$MSE(y, p) = \frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2 \tag{3.13}$$

Where y_i is the true value and p_i is the predicted value of the i-th sample.

Classification Accuracy : For the Classification task, the main quality indicator used, is the percentage of accurately predicted cases. For example, if the dataset contains two classes and fifty percent of the records is assigned in the one class, then an accuracy of fifty percent means, that our prediction is as good as random. While an accuracy, of a hundred percent means a perfect prediction. This measurement is simple and can be misleading in cases where the dataset is skewed, like in the case where we do not have the same number of samples for all classes.

Area Under Curve & Receiver Operating Characteristic : A more precise quality indicator can be obtained with the use of the *Receiver Operating Characteristic*, ROC, curve. This curve shows the trade-off between correctly classified records of the "positive" class, versus incorrectly classified records as the positive class. Using the probabilities predicted by a classifier this curve can be calculated, in order to give an indication of the overall precision of the classifier. Since the ROC curve is a two dimensional curve, it is difficult to compare different classifiers based on this metric. Due to this fact, the *Area Under the Curve*, AUC, is commonly used to measure the propriety of the predictor. By the optimization of the AUC score, we gain the optimization of the predictor. The optimal value of the AUC metric is one, while 0.5 means that the predictor predicts randomly.

Chapter 4

Results

In the following section, the results from the regression as well as for the classification task are shown. Following those results, are described additional experiments, in Section 4.3. Finally, in Section 4.4, the results of our data exploration are shown and discussed.

4.1 Regression Results

Gritsenko

For the Gritsenko dataset, many different regression algorithms were used on the data, in order to see the prediction results that could be retrieved. A wide variety of parameters were explored for those models. However, only the empirically best parameters are presented in the tables below.

For all the experiments we have tried on Gritsenko's dataset, we apply two different scenarios. At first we apply all the features that were generated in our research and those are: the length, A's, T's, G's, C's frequencies, Dimer's and Trimer's frequencies as well Stress and MFE. And as a second experiment include four additional features, the values of which were used by Gritsenko et al. in order to derive the translation initiation rates in their research [8]. Those extra four features are: the average mRNA reads, the average Ribosome reads, the Fitness and the number of Segments per gene. The reason we include them in our research is in order to validate the propriety of our experimental approach.

The coefficient of determination, R^2 score, for both the test and validation data are shown in the tables below.

TABLE 4.1: Results per fold on Gritsenko's data [8] using all features **excluding** the following four features: Avg. norm. mRNA read count, Avg. norm. Ribosome read count, Fitness, Number of Segments

Algorithm	Fold	Test R^2	Validation R^2	Parameters
Random Forest	1	0.01138582	0.0216924	n_estimators=100
Random Forest	2	0.03516566	0.04285264	n_estimators=100
Random Forest	3	0.03906075	0.04998809	n_estimators=100
Random Forest	4	0.0342417	0.0076032	n_estimators=100
Random Forest	5	0.02319767	0.05236511	n_estimators=100
Regression Tree	1	-0.03375269	0.02029835	Max_depth=4
Regression Tree	2	0.00163356	-0.04590142	Max_depth=4
Regression Tree	3	-0.06398509	-0.03655098	Max_depth=4
Regression Tree	4	-0.00696978	-0.00525878	Max_depth=4
Regression Tree	5	-0.04896083	-0.02816244	Max_depth=4
Kriging	1	0.00917133	-0.01011966	Default
Kriging	2	0.01825115	-0.00742969	Default
Kriging	3	0.00511215	-0.01858246	Default
Kriging	4	0.01193798	-0.00499815	Default
Kriging	5	-0.01039905	-0.01619794	Default
SVM	1	0.01981213	-0.07325528	Default
SVM	2	-0.02912333	-0.05892615	Default
SVM	3	0.03677134	-0.08058316	Default
SVM	4	-0.01363057	-0.05508755	Default
SVM	5	-0.00783102	-0.05535512	Default
LASSO	1	-0.00483726	-0.00073398	Default
LASSO	2	-0.00147557	-0.00229592	Default
LASSO	3	-0.00518079	-0.00070988	Default
LASSO	4	-0.00831358	-0.00332039	Default
LASSO	5	-0.00018453	-0.00186482	Default

TABLE 4.2: Results per fold on Gritsenko's data [8] using all features **including** the following four features: Avg. norm. mRNA read count, Avg. norm. Ribosome read count, Fitness, Number of Segments

Algorithm	Fold	Test R^2	Validation R^2	Parameters
Random Forest	1	0.82954783	0.84565613	n_estimators=100
Random Forest	2	0.81386788	0.84247368	n_estimators=100
Random Forest	3	0.84122053	0.83436637	n_estimators=100
Random Forest	4	0.83710895	0.84661551	n_estimators=100
Random Forest	5	0.85066029	0.83289239	n_estimators=100
Regression Tree	1	0.63966753	0.61078938	Max_depth=4
Regression Tree	2	0.57592482	0.61693205	Max_depth=4
Regression Tree	3	0.5905166	0.6157789	Max_depth=4
Regression Tree	4	0.62258875	0.60604124	Max_depth=4
Regression Tree	5	0.68141824	0.65150359	Max_depth=4
Kriging	1	0.18656065	0.27656157	Default
Kriging	2	0.25894087	0.26877001	Default
Kriging	3	0.2381838	0.29261151	Default
Kriging	4	0.17664955	0.27367013	Default
Kriging	5	0.24141091	0.29874874	Default
SVM	1	0.30533241	0.34264988	Default
SVM	2	0.33272437	0.36323715	Default
SVM	3	0.20991707	0.33277265	Default
SVM	4	0.31840021	0.33531972	Default
SVM	5	0.26967266	0.33360603	Default
LASSO	1	0.24298908	0.28598520	Default
LASSO	2	0.28090668	0.28560219	Default
LASSO	3	0.30414779	0.28491968	Default
LASSO	4	0.32140461	0.28447393	Default
LASSO	5	0.26054854	0.28749306	Default

TABLE 4.3: Succinct results Gritsenko's data [8] for the features Sequence Length, Stress, MFE, A's,T's,G's,C's frequencies, Dimers's and Trimers's frequencies.

Algorithm	AVG Test R^2	Test Std.	Val. Avg. R^2	Val. Std.	MAE train	MAE Test
Random forest	0.03	0.01	0.03	0.01	0.3000	0.7587
Regression Tree	0.03	0.02	0.02	0.02	0.7772	0.8003
Kriging	0.01	0.01	0.01	0.01	1.8836e-15	0.8136
SVM	0.00	0.02	0.06	0.01	0.4919	0.8018
LASSO	-0.003	0.00	-0.002	0.00	0.8062	0.8391

TABLE 4.4: Succinct results Gritsenko's data [8] using the above features and including the four following features: Avg. norm. mRNA read count, Avg. norm. Ribosome read count, Fitness, Number of Segments

Algorithm	AVG Test R^2	Test Std.	Val. Avg. R^2	Val. Std.	MAE train	MAE Test
Random forest	0.83	0.01	0.84	0.01	0.1158	0.2964
Regression Tree	0.62	0.04	0.62	0.02	0.4491	0.4394
Kriging	0.22	0.03	0.28	0.01	4.0710e-15	0.6820
SVM	0.29	0.04	0.34	0.01	0.3630	0.6909
LASSO	0.28	0.03	0.28	0.00	0.6780	0.6615

From Table 4.1, it can be observed that for all models and parameter combinations, both test and validation score are quite not satisfactory. Generally, we can observe that the Random Forest algorithm, seems to have the best performance. In order to evaluate that our approach is sound, we repeated the experiments including four additional features, that we know for sure that the initiation rates are depending on them. These additional features are: Average mRNA Read, Average Ribosome Reads, Fitness and the Number of Segments. As it would be expected, in the Table 4.2 one can see that the R^2 scores are very much improved in relation to the previous results. Unfortunately, the prediction of the translation initiation rates seems to be very hard, if not impossible, when these features are not present. The Random Forest model shows the most promising results in the executed experiments, with an R^2 score of up to 0.84 including the four additional features and an R^2 score of 0.03 excluding these features.

Ciandrini

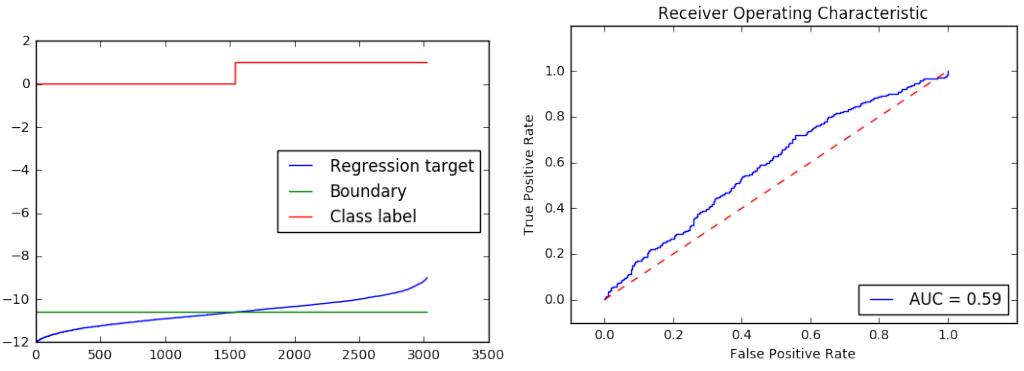
For the Ciandrini dataset, the same experiment is repeated using the same models and parameters.

From Table 4.5 it can be observed, that also predicting the initiation rates for this dataset seems to be a very hard task. Similarly to the Gritsenko experiments, Random Forests seem to be the most promising model to use for the Ciandrini data too.

Since predicting the exact initiation rates is shown to be a very hard task, perhaps predicting the extreme cases can be done instead. In order to check whether we can obtain, at least, an indication of the extreme (low and high) initiation rates, as a next step, is performed the task of predicting whether the target is low or high. Turning, thus, the problem into a Classification task.

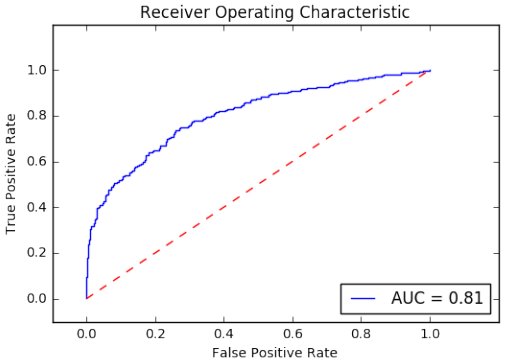
4.2 Classification Results

The chosen classification model, is the *Gradient Boosting Classifier*. This model is chosen by *Auto-sklearn* [37], after the fine tuning of various models and parameters.

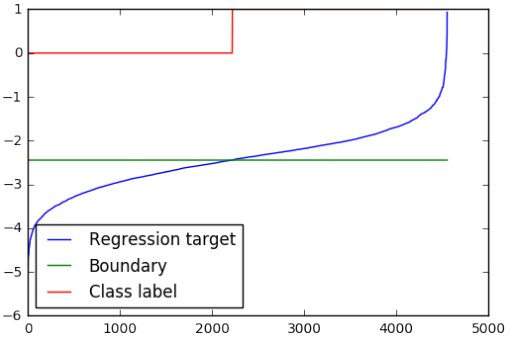


(A)

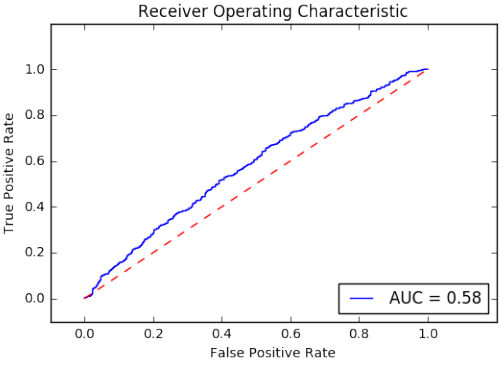
(B)



(C)



(D)



(E)

FIGURE 4.1: (A) Preprocessing of the regression problem into a classification problem by applying a binary class label to the records. Here is shown the Gritsenko [8] dataset without the inclusion of the four extra features. Target values below the mean are assigned class label-0 and values above the mean are assigned class label-1. (B) ROC curve calculated for the Gritsenko [8] dataset, with the use of *autosklearn.classification*[37] and more specifically the Gradient Boosting Classifier. (C) ROC curve calculated for the Gritsenko [8] dataset, with the use of *autosklearn.classification*[37]. The difference with (B), is that in this case, there were added four extra features. Those features are the: Average mRNA Reads, Average Ribosome Reads, Fitness and Number of Segments. (D) Similar to (A) and (E) is similar to (B) but in this case, they are calculated for the Ciandrini [11] dataset.

TABLE 4.5: Results per fold on Ciandrini's data [11] for the features Sequence Length, Stress, MFE, A's,T's,G's,C's frequencies, Dimers's and Trimers's frequencies.

Algorithm	Fold	Test R^2	Validation R^2	Parameters
Random Forest	1	0.04317389	0.0008724	n_estimators=100
Random Forest	2	0.00188718	0.01239863	n_estimators=100
Random Forest	3	0.02114724	0.0122576	n_estimators=100
Random Forest	4	0.00606485	0.01120197	n_estimators=100
Random Forest	5	0.02844071	-0.00136179	n_estimators=100
Regression Tree	1	-0.01924838	0.00653142	Max_depth=4
Regression Tree	2	0.08662839	0.02060814	Max_depth=4
Regression Tree	3	-0.03605583	-0.09009538	Max_depth=4
Regression Tree	4	-0.03914204	-0.00059705	Max_depth=4
Regression Tree	5	-0.00720636	-0.00761121	Max_depth=4
Kriging	1	0.01654896	0.01535292	Default
Kriging	2	0.01149844	0.01925918	Default
Kriging	3	0.01865919	0.02440714	Default
Kriging	4	0.00160740	0.02440714	Default
Kriging	5	0.00160740	0.03156701	Default
SVM	1	-0.06814349	-0.05674336	Default
SVM	2	-0.02694341	-0.04992307	Default
SVM	3	-0.00599307	-0.06808839	Default
SVM	4	-0.0274114	-0.02092128	Default
SVM	5	-0.03866553	-0.06018205	Default
LASSO	1	-0.00267240	0.00085043	Default
LASSO	2	0.000755587	0.00107102	Default
LASSO	3	0.000181063	-0.0013816	Default
LASSO	4	0.000972333	0.00033517	Default
LASSO	5	-0.00351892	-0.0026633	Default

The classifier produces the ROC curves, that are shown in the figures 4.1e, 4.1b, 4.1c, after fitting on the data. In this case, there were two different kinds of experiments performed. In the first, the dataset is split into two classes using the mean initiation rate as boundary. This is shown in Figures 4.1a and 4.1d. What can be seen in those figures is the log values of the translation initiation rates sorted, depicted with blue color. The green line shows the mean value of our target. And with the red color can be seen the transformed classification target. All the initiation rates with a value below that of the mean, is assigned to the class label-0 and those with a higher value, are assigned to label-1. In the Figures 4.1b, 4.1e, one can see the ROC curve calculated for the Gritsenko and Ciandrini data sets, respectively. In those figures, the red dashed line shows a base line random classifier. The blue line shows the ROC curve of the Gradient Boosting Classifier. The area under curve score, AUC, is given in the legend. It can be observed that for both data sets, the calculated ROC is very close to the base line. We can make a degree of distinction between the low and high initiation rates, but there are still many false positives. Similarly to previous experiments, in order to validate whether our results are correct, we have included the extra four features for the Gritsenko database. The results of this experiment can be seen in the Figure 4.1c, where it can be observed that the AUC score has been improved from 0.59 to 0.81.

In the second experiment, only the extreme values for the translation initiation rates were taken into account. In Figures 4.2a and 4.2c we can again see the sorted logarithmic values of the translation initiation rates depicted with the blue line. The red line shows the upper threshold,

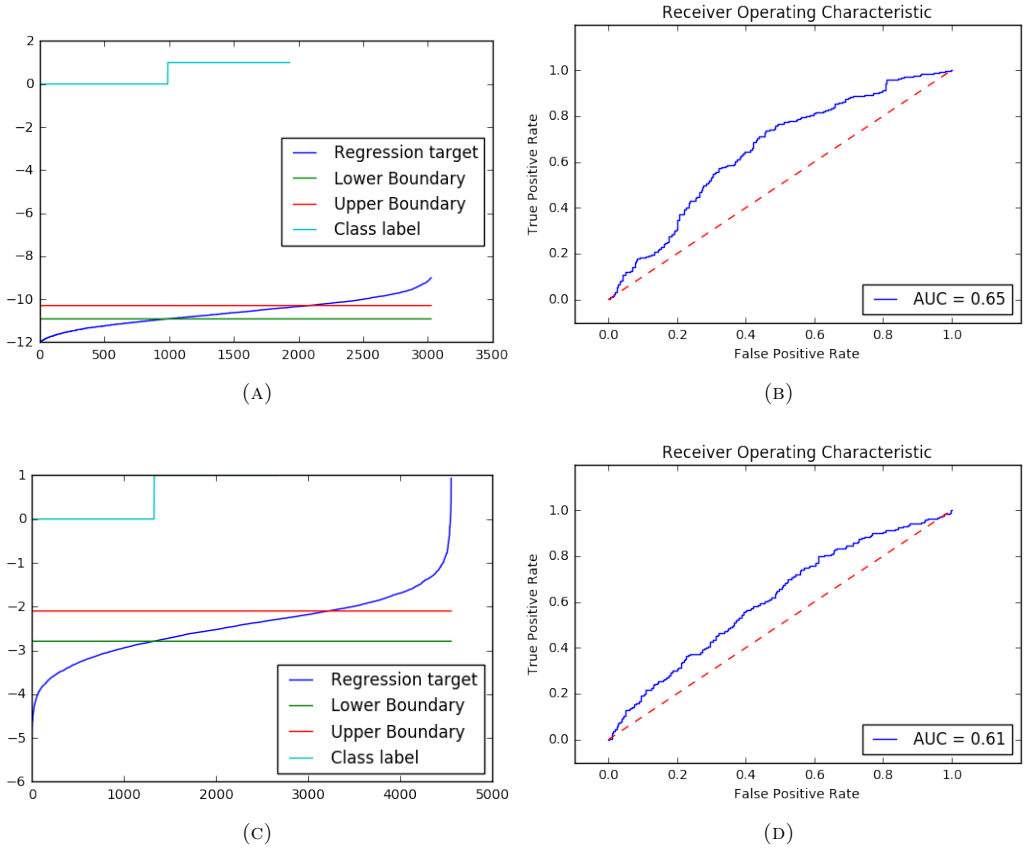


FIGURE 4.2: (A) Preprocessing of the regression problem into a classification problem but now by removing the targets around the mean value, taking only the extreme cases into account. (B) ROC curve calculated using only the dataset with the extremes as visualized on the left for the Gritsenko dataset. (C) Similar as (A), but this time for the Ciandrini dataset. Respectively, (D) is similar to (B).

which is set to $\mu + \frac{\sigma}{2}$. The lower threshold, depicted by the green line, is set to $\mu - \frac{\sigma}{2}$. Samples with initiation rates below the lower threshold are assigned to label-0, whereas those with values higher than the upper threshold are assigned to label-1. The samples with values in between, are omitted. The ROC curves of this second type of experiment are shown in Figures 4.2b and 4.2d for the Gritsenko and Ciandrini dataset respectively. One would observe, in those figures, that the classification accuracy is slightly improved for both data sets. Especially, for the Gritsenko dataset, an improvement can be clearly observed. Concluding thus, that extreme initiation rates can be easier distinguished.

4.3 Additional Experiments

As an additional attempt to increase the performance of the classifiers, *Principal Component Analysis*, (PCA) is used to see whether reducing the dimensionality of the data sets can improve the performance. In Figures 4.3a, 4.3b, 4.3c and 4.3d, the AUC score of the trained predictor is plotted on the y axis against the number of PCA components on the x axis. *Principal Component Analysis*, is a statistical procedure that is used to emphasize variation and bring out the strong

patterns in a dataset. Is like one is willing to take a photo of our data, and tries to find the best angle to make this photo, so that he can retrieve enough information about the nature of the data, without taking photos from each side of them. Thus, PCA is used as a dimensionality reduction method, selecting the features that best describe our data.

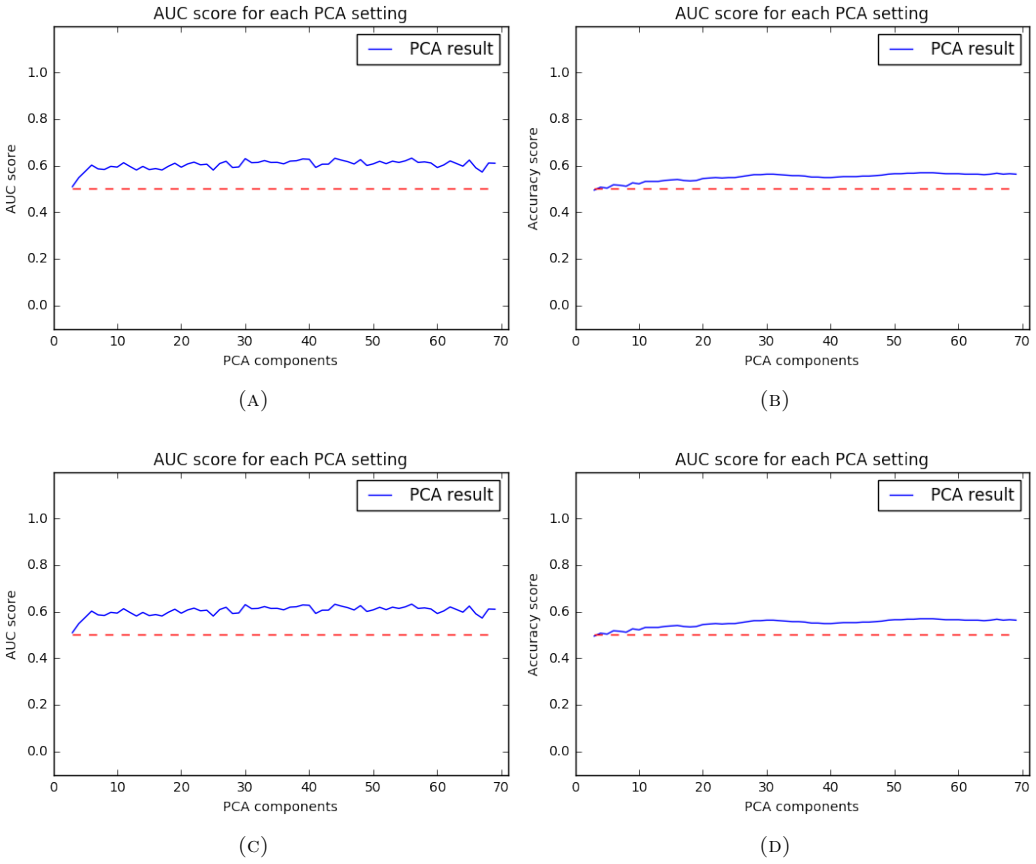


FIGURE 4.3: (A) AUC scores for different PCA preprocessing settings using a Gradient Boosting Classifier for the Gritsenko dataset. (B) Similar to (A), but in this case using the Support Vector Classifier. (C) AUC scores for different PCA preprocessing settings using a Gradient Boosting Classifier for the Ciandrini dataset. (D) Again the AUC score for the Ciandrini data, but this time using the Support Vector Classifier.)

It can be observed from the Figures 4.3a and 4.3c, that no matter the number of the principal components included in the experiment, the performance remains almost the same. However, for the Support Vector CLassifier the accuracy is slightly increasing when the number of principal components increases. This can be seen in Figures 4.3b and 4.3d. Something more that we could observe, is that the Gradient Boosting Classifier performs slightly better than Support Vector Classifier. Unfortunately, the use of PCA in our experiments does not improve the performance opposed to our aforementioned results.

4.4 Data Exploration

In order to further investigate why the regression and classification results from the experiments are so low, extensive data exploration is performed. In this section, using various data exploration

methods explained in Section 3.3, we reason about the separability of the translation initiation rates. That is because we would like to explain the difficulty of the prediction task for data resulting from stochastic models.

The figures referenced in this section, can be found in Chapter 6: Appendix A. In the Figures 6.1 and 6.2, is shown the feature correlation between the different features and the target. From Figure 6.2 one can observe that, as expected, the features *Average mRNA Reads*, *Average Ribosome Reads*, *Fitness* and *Number of Segments* show high correlation with the target. Unfortunately, none of the other features seem to show any linear correlation with the target. Which is surprising since, at least a correlation between the target and the length of the 5'UTR was expected, according to Ciandrini et al [11].

In Figures 6.7 and 6.8 are shown the feature importances, as calculated by the random forest regressor trained on the Gritsenko data set. The feature importances calculated from a random forest model show the same pattern as the correlations mentioned before, while ATG seems to be the most important feature in Figure 6.7. Figure 6.8 shows to us, that by including the four additional features the feature importances change dramatically.

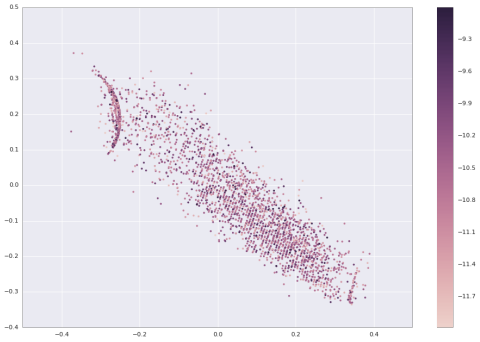
While having a look at the features, that we have gathered, and the translation initiation rates, from a different point of view, we have plot them in a clever fashion. Especially since both the Gritsenko and the Ciandrini data sets, have more than 3000 records and more than 90 features. In order to better visualize this data, we can make use of several plotting algorithms for high dimensional data sets.

In the Figures 4.4a, 4.4b and 4.4c, with darker color are shown the sequences with high translation initiation rates, and with lighter color, those with lower rates. If we have a close look into the Distance Matrix for the Ciandrini dataset 4.4a, we can see that the high and the low values are scattered. It is very hard to divide the data into low and high initiation rates regions. Similar behavior is seen in the first image for the Gritsenko dataset 4.4b. However, having a closer look into the Figure 4.4c, we can observe a gathering of the genes with high initiation rates at the lower region and those with lower values in the upper region. We can almost say that we can draw the boundary line, that separates the data into those of high and low translation initiation rates. The difference between the two Figures for the Gritsenko data, is that in the second case the four additional features were added.

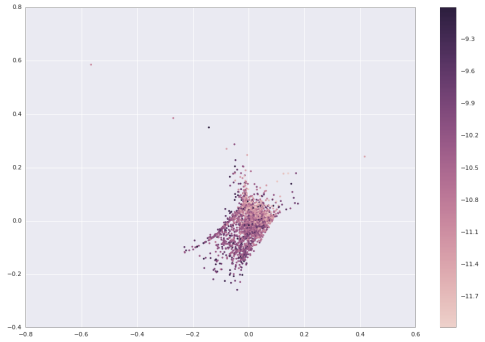
Lag plots are a nice way to inspect whether the collected data is random or not. If the data is random, then the Lag plot shows no clear pattern. The Lag plots of the Gritsenko and Ciandrini dataset are shown in Figures 4.5a and 4.5b. It can be observed that the plot is not completely random. However, both datasets show a large area with seemingly random behavior.

The *Andrews' Curves* for both data sets are shown in Figures 4.6a and 4.6b. It can be observed, that for both data sets similar behavior occurs, and that both high and low translation initiation rates are very much overlapping. In the case of the *Andrews' Curves*, one can observe that there is a slight difference among the two classes. For example, the class one is located higher than the class zero. The colors in those figures, where randomly assigned by the algorithm, so they are not matching among the two figures. But, we can still observe similar behavior between the

(A)

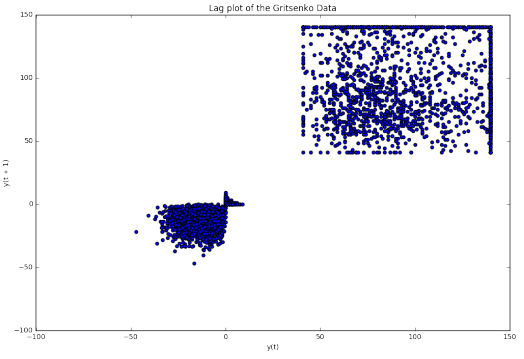


(B)

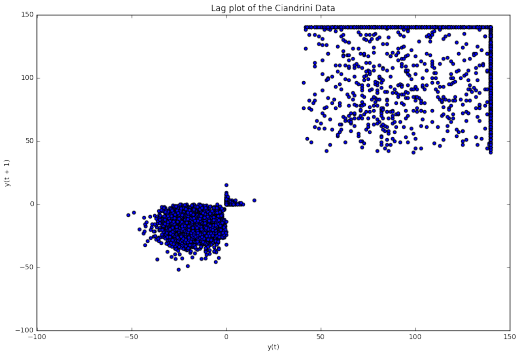


(C)

FIGURE 4.4: (A) The Distance Matrix calculated for the Ciandrini [11] dataset[37] with the use of the Manhattan distance. (B) The Distance Matrix calculated for the Gritsenko [8] dataset, with the use of the Manhattan distance. (C) The Distance Matrix calculated for the Gritsenko [8] dataset. But in this case, are included the extra features: Average mRNA Reads, Average Ribosome Reads, Fitness and Number of Segments.



(A)

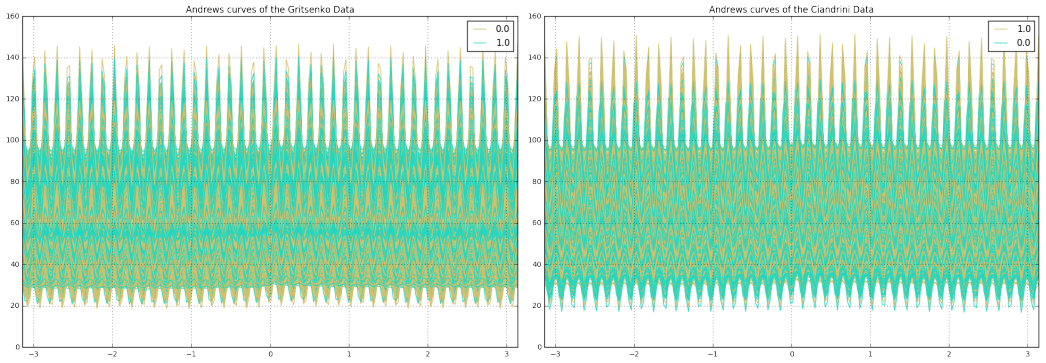


(B)

FIGURE 4.5: (A) Lag Plot of the Gritsenko dataset without the extra four features. (B) Lag Plot of the Ciandrini dataset.

two data sets. For instance, we can observe that in both the Gritsenko data and the Ciandrini data, the class one is located higher than the class zero.

RadViz plots plot the records depending on the feature values, each feature becomes a dot

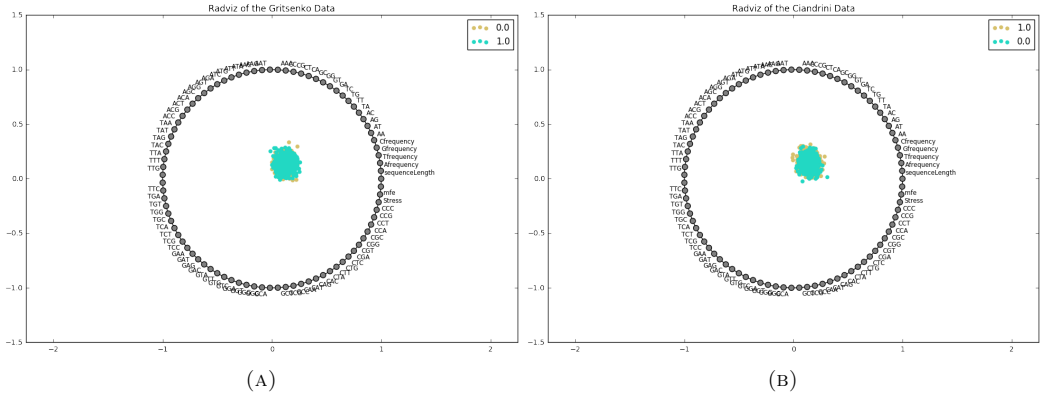


(A) (B)

FIGURE 4.6: (A) Andrews' Curves of the Gritsenko dataset without the extra four features and (B) for the Ciandrini dataset.

uniformly distributed in a circle. See Figures 4.7a and 4.7b. There are invisible springs between the feature dot on the circle and the dot that represents the record, if the record has a relative high value for a certain feature, then the tension in the spring is high. Using these tensions, the position of the record dot is calculated. In that way, is easier to verify, whether there are different clusters in the data.

To visualize the *Andrews' Curves* and *RadViz plots* nicely, we can use a class label, to color the different records. Here we choose to assign two classes to our data sets, 1 and 0 for high initiation rates and low initiation rates respectively. We do this by looking only at the more extreme cases in the dataset. All records with a translation initiation rate higher than the mean plus half standard deviation, $(\mu + \frac{\sigma}{2})$, of the target, receive the class label 1. Similarly, all the records, with a target below the mean minus the half standard deviation, $(\mu - \frac{\sigma}{2})$, receive the class label 0. The remaining records were discarded for the sake of those visualizations.



(A) (B)

FIGURE 4.7: (A) RadViz Plot of the Gritsenko dataset without the extra four features. (B) RadViz Plot of the Ciandrini dataset.

From these plots, it can be concluded that predicting exact or distinguishing between low and high translation initiation rates using the selected features is a very difficult task. The data seem to be not separable in the feature space and even extreme cases are highly overlapping.

Conditional Entropy

During data exploration, we were interested to see the impact of specific nucleotides on the translation rates. For that reason we calculated the *Conditional Entropy*. The conditional entropy measures how much entropy, which is the lack of predictability, a random variable X has; remaining if we have already learned the value of a second random variable Y. It is referred to as the entropy of X conditional on Y, and is written $H(X|Y)$. In our case the random variables are the four nucleotides: A,T,G and C. What we are aiming with this experiment is to find a pattern of nucleotides that occurs at a specific position among the 10% of the sequences with the highest translation initiation rates.

The Conditional Entropy is calculated by the following formula:

$$E_{set[i,k]} = P_{set[i,k]} * log_2(P_{set[i,k]}/P_{back[i,k]}) \tag{4.1}$$

Where $P_{set[i,k]}$ denotes the probability of a nucleotide i at position k in the 10% of the sequences with the highest translation initiation rates. $P_{back[i,k]}$ is the probability of the same nucleotide at the same position in the remaining 90% of the sequences. The relative height of the nucleotide bars equals to $E_{set[i,k]}$. When this value is positive, leads to enrichment and when negative to depletion, according to Dvir et al. [3]. At first we used the shortened sequences with length as defined in our features, Section 3.2, where we cut out the 5' UTR area above 100bp and we do use of the first 40bp of coding sequence, and thus we have sequences of maximum 140bp length. For the calculation of the conditional entropy, is important to align the sequences, in order to be sure that a specific position is common across all data. However, not all sequences have the same length in their 5' UTR. Thus, we have reversed all the sequences in such a way that the 40th base of coding sequence becomes the first base of the sequence, and the last base of 5' UTR is the end. This can be more clearly seen in the Figure 4.8. What we could conclude from this figure is that 15% among the 10% of the sequences with the highest initiation rates have a nucleotide A at 129bp position.

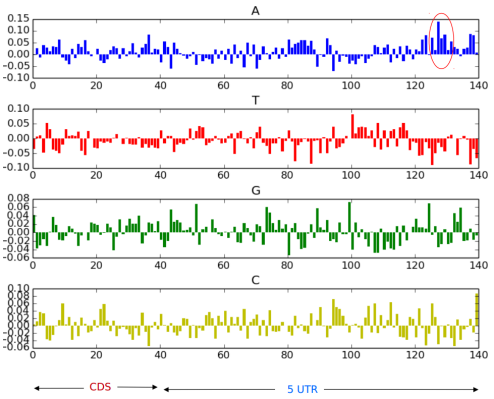


FIGURE 4.8: The Conditional Entropy calculated for the Gritsenko dataset. The hight of the bars denote the $E_{set[i,k]}$ value for each base A,T,G and C. Here, were used only the last 100bp of the 5'UTR sequence. Note that for alignment reasons we have reversed the sequences starting from the 40 first bp of coding sequence and moving on towards the 5' UTR.

The same experiment we tried for all the 3-mers as well (AAA, ATA, ..., CCC). In that case we were interested to see whether a combination of three bases at a specific position can give us some distinguishing information for the sequences with the highest initiation rates. Figure 4.9.

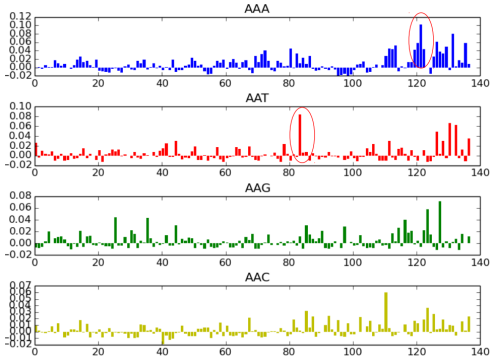


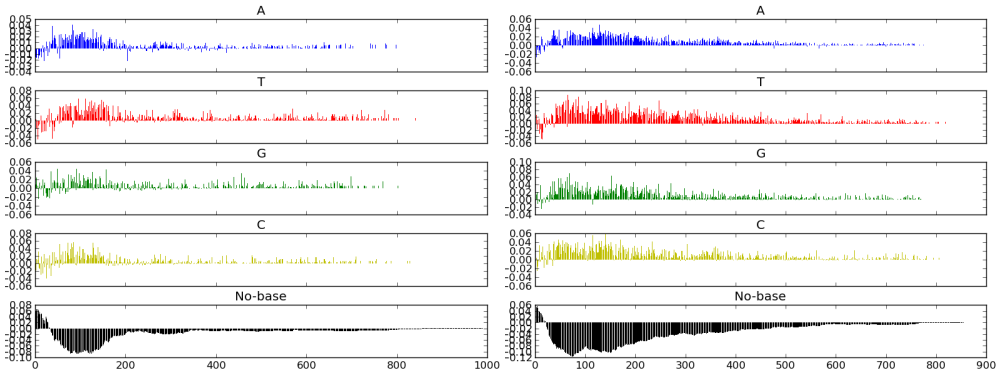
FIGURE 4.9: The Conditional Entropy calculated for the Gritsenko dataset for the trimers. Here is shown only the first figure of the results. The hight of the bars denoete the $E_{set[i,k]}$ value for each trimer eg AAA, ATA, ... , CCC. Here, were used only the last 100bp of the 5'UTR sequence. Note that for alignment reasons we have reversed the sequences starting from the 40 first bp of coding sequence and moving on towards the 5' UTR.

As a following step, we considered more useful to use the whole length from the 5' UTR sequence. We were also willing to include the calculation of the conditional entropy in our regression algorithm. In order to achieve this, we summed up the $E_{set[i,k]}$ values, for all the nucleotides per sequence, so that every sequence has a score. Thus, we have used this score as another feature in our prediction model. There was a small normalization problem though. Short and long sequences might have the same exact score. In order to handle this, we included into our calculation the spaces as well, that is referred as *No-base* in Figure 4.10. The results of the regression experiment are not included in this research, as they were inferior to our final results.

Having a better look at the figure 4.10, one could observe, that sequences with almost no 5' UTR sequence seem to be translated more often. We can see that, because short sequences, that are sequences that have positive values for No-base in the first 40bp are among the 10% of sequences with the high initiation rates. This verifies our initial assumption about the negative correlation between the length and the translation initiation rates. Similarly, we can observe an increase of the CE values for No-base after 200bp too. Which means that sequences with very long 5' UTR are also among the 10% of the sequences with high initiation rates.

Sliding Window

In order to maintain the position dependent information across the sequences and to see whether certain areas are more informative than the whole sequence, we performed a *Sliding Window* approach, for the prediction of the translation initiation rates. In other words, we divided every sequence in sub-sequences. Moving from the 5' prime end towards the 3' prime end, we generated small sequences of 10 base pairs each. We moved along the sequence with 1bp step, which means 9bp overlap between any two consecutive sub-sequences. Our initial sequences were of 140bp maximum length, which led to 130 windows ($n - windowsize$). A schematic representation of the sliding window can be seen in Figure 4.11.



(A) (B)

FIGURE 4.10: (A) The Conditional Entropy calculated for the Gritsenko dataset across the whole length of every sequence. (B) Similarly, the conditional entropy calculated for the Ciandrini dataset. On the x axis we have the length of the sequences in bp, whereas on the y axis the values of the $E_{set[i,k]}$. Note that for alignment reasons we have reversed the sequences starting from the 40 first bp of coding sequence and moving on towards the 5' UTR. In this Figure it can be seen that sequences with negligible 5' URT sequence or 5' UTR length longer than 160bp are among the 10% of the sequences with the highest initiation rates.

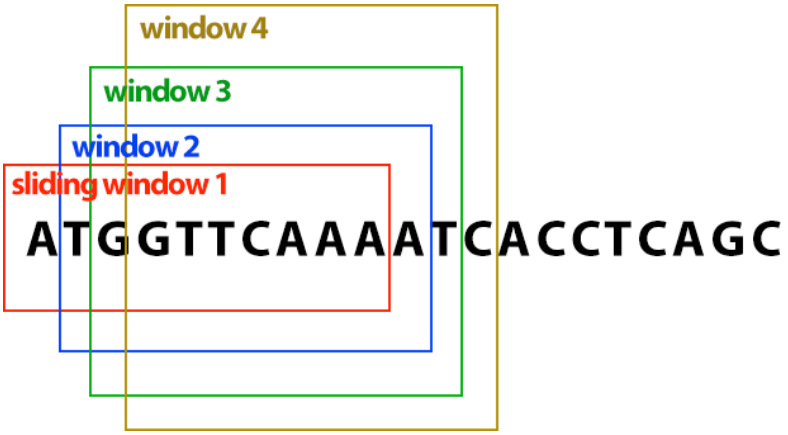


FIGURE 4.11: sliding window

For every window, we have generated separately, all the aforementioned features: A's, T's, G's, C's frequencies, Dimer's and Trimer's frequencies. In this manner, we have increased the dimensionality of our problem. After the generation of the features, we tried the following experiments:

- Calculate the feature correlation against the translation initiation rates. (Pearson and Spearman correlation). As well as the feature importances calculated by the Random Forest regressor [27].
- Performed a double loop 3-fold cross validation for the regression. For the regression, the following models were evaluated: Random Forest Regressor, Gaussian Process Regression, Support Vector Regressor, LASSO.
- Applied a Prinicipal Components Analysis using 20, 40 and 80 components as an input to a Random Forest Regressor.

Following tables present the results of the sliding window for the two datasets using random forest regressor 4.6, 4.7, 4.8. All the other results related to the sliding window were omitted as they were inferior. The features that we feed the regressor with are the MFE and Stress features per gene as well as the A's, T's, G's, C's frequencies, Dimer's and Trimer's frequencies calculated for each window. The correlation images, after applying the sliding window, can be found in 6.4, 6.5 and 6.6. Additionally the images for the feature importance as they arise from the random forest can be found in 6.10, 6.11 and 6.12.

TABLE 4.6: Results per fold on Gritsenko's data [8] using all features as they arise from the sliding window, **including** the following four features: Avg. norm. mRNA read count, Avg. norm. Ribosome read count, Fitness, Number of Segments

Algorithm	Fold	Test R^2	Validation R^2	Parameters
Random Forest	1	0.76856104	0.76271527	n_estimators=100
Random Forest	2	0.76481020	0.78313925	n_estimators=100
Random Forest	3	0.79345205	0.76424642	n_estimators=100
Random Forest	4	0.80199309	0.75221641	n_estimators=100
Random Forest	5	0.76589967	0.75577879	n_estimators=100

TABLE 4.7: Results per fold on Gritsenko's data [8] using all features as they arise from the sliding window, **excluding** the following four features: Avg. norm. mRNA read count, Avg. norm. Ribosome read count, Fitness, Number of Segments

Algorithm	Fold	Test R^2	Validation R^2	Parameters
Random Forest	1	0.02705442	0.02916387	n_estimators=100
Random Forest	2	0.03368827	0.03514938	n_estimators=100
Random Forest	3	0.03906814	0.01492746	n_estimators=100
Random Forest	4	0.00588796	0.00971480	n_estimators=100
Random Forest	5	0.01972022	0.01262269	n_estimators=100

TABLE 4.8: Results per fold on Ciandrini's data [11] using all features as they arise from the sliding window.

Algorithm	Fold	Test R^2	Validation R^2	Parameters
Random Forest	1	0.01477579	0.00822615	n_estimators=100
Random Forest	2	0.01011842	0.00080927	n_estimators=100
Random Forest	3	0.0043851	0.01592642	n_estimators=100
Random Forest	4	0.03176115	-0.01087973	n_estimators=100
Random Forest	5	0.01578811	0.02916065	n_estimators=100

Chapter 5

Conclusions and Future Research

5.1 Conclusions

Having a more careful look at the results 4.3 & 4.4, one can observe that the scores of all the chosen methodologies are significantly better with the inclusion of the features Average mRNA reads, Average Ribosome reads, Fitness and the Number of Segments. The values of which, have been used for the derivation of the Translation Initiation Rates in the paper Gritsenko et al. [8]. This is quite expected and proves that our algorithms actually work. Unfortunately, this is also a confirmation that none of the rest of the features happens to be a significant opposer, since the results from the regression task show results similar to that of a random predictor.

Predicting the exact value of our target, the translation initiation rates, seems to be a hard task if not impossible. Most likely due to the fact that the initiation rate values are produced by stochastic models and only partly reflect the real initiation rates in nature. As a next step we tried to predict, whether the target is either low or high. This is done by transforming our regression task into a classification task. The results of these classification experiments (Figures 4.1 and 4.2) show us, that this is indeed an easier task than predicting the exact value. However, as it can be observed from the ROC curves in Section 4.2, the AUC score of the classification task for both data sets is around 0.6, which is not a very confident score either. When including the additional features, this AUC score and the accuracy of the predictor goes up to 0.8, showing again that by using these extra features, the prediction can be made much more reliable. From the feature importances for both data sets we can see, however, that the *MFE* seems to play a role in relation with the translation initiation rates. Unfortunately, it is not clear from those results on how the MFE influences the initiation rates.

From the *Principal Component Analysis* experiment, that is shown in Figure 4.3, it can be observed, that dimensionality reduction does not increase the performance of the classifier. Similarly, for the optimized Gradient Boosting Classifier and the Support Vector Classifier, the results show the same quality and there is no increase of the performance observed.

As it occurs from the Conditional Entropy calculation in our additional experiments, the length of the 5'UTR sequence seems to play a role in the translation frequency of the sequences as well.

Is quite interesting to observe in Figure 4.10, that sequences with negligible 5'UTR are among the 10% of the sequences with the highest translation initiation rates. In the same figure, it can also be observed that sequences with 5'UTR sequence longer than 160bp, seem to be translated more frequently than those with smaller.

Lastly, having an additional look in the Correlation Figures 6.1 and 6.3, one can see that the MFE feature seems to have indeed higher correlation with the target, than other features, as it has been suggested by Ciandrini et al. [11]. However, this correlation is 0.12 for the Ciandrini data and 0.06 in the Gritsenko data, and thus is considered insignificant. Additionally, from the image depicting the feature importances, Figure 6.9, we can again see that the MFE seems to play a significant role in the Ciandrini data, as it has been mentioned in the relative literature. Whereas in the Gritsenko's dataset, Figure 6.7, the MFE comes in the third important position. Subsequently, we can come to the conclusion, that the two datasets do differ, although there are many similarities among them.

To sum up, the predicting task of the translation initiation rates out of the 5'UTR sequences seems not to be a trivial task. The reason to this might be the fact that the data we have used are products of stochastic models and not actual data [8, 11]. That is because stochastic model's data contains noise.

5.2 Future Research

There are still many open questions and possibilities to explore in this area. Additional features could be used in the predictive algorithms. An example of such an additional feature would be the presence of a specific pattern in a sequence. Such a pattern could be the consensus of the *Ribosomal Binding Site*, which is the AGGAGG. Another interesting feature would be the investigation of the exact position of such a sequence.

An interesting research area would be the application of motif discovery on sequences with a high initiation rate, in order to see whether there are specific motifs that might contribute to the initiation rate. This is something we have already tried for small motifs of three base pairs. Thus as a future experiment we could try longer motifs.

As a next step one could try different predictive models as well. Perhaps the use of Deep Neural Networks would be of an advantage since the construction of features would be automated. This could lead to the improvement of the prediction performance.

Acknowledgments

E.T. would like to thank M.J.T. Reinders for his guidance and critical eye during the completion of this research. Would also like to thank A. Gritsenko for his patience and his cornucopia of ideas. Lastly, I would like to thank my husband Bas van Stein, without his support this research would have never been completed.

Bibliography

[1] Francis Crick et al. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

[2] Wikipedia, template:eukaryote gene structure. URL https://en.wikipedia.org/wiki/Template:Eukaryote_gene_structure.

[3] Shlomi Dvir, Lars Velten, Eilon Sharon, Danny Zeevi, Lucas B Carey, Adina Weinberger, and Eran Segal. Deciphering the rules by which 5' -UTR sequences affect protein expression in yeast. 2013. doi: 10.1073/pnas.1222534110/-/DCSupplemental.www.pnas.org/cgi/doi/10.1073/pnas.1222534110.

[4] Markus Ringnér and Morten Krogh. Folding free energies of 5'-UTRs impact post-transcriptional regulation on a genomic scale in yeast. *PLoS Computational Biology*, 1 (7):0585–0592, 2005. ISSN 1553734X. doi: 10.1371/journal.pcbi.0010072.eor.

[5] Tao Huang, Sibao Wan, Zhongping Xu, Yufang Zheng, Kai Yan Feng, Hai Peng Li, Xiangyin Kong, and Yu Dong Cai. Analysis and prediction of translation rate based on sequence and functional features of the mRNA. *PLoS ONE*, 6(1):4–11, 2011. ISSN 19326203. doi: 10.1371/journal.pone.0016036.

[6] Benjamin Reeve, Thomas Hargest, Charlie Gilbert, and Tom Ellis. Predicting Translation Initiation Rates for Designing Synthetic Biology. *Frontiers in Bioengineering and Biotechnology*, 2(January):1–6, 2014. ISSN 2296-4185. doi: 10.3389/fbioe.2014.00001. URL http://www.frontiersin.org/Synthetic_Biology/10.3389/fbioe.2014.00001/abstract.

[7] David R Morris and Adam P Geballe. *Molecular and cellular biology*. doi: 10.1128/MCB.20.23.8635-8642.2000.Updated.

[8] Alexey A. Gritsenko, Marc Hulsman, Marcel J T Reinders, and Dick de Ridder. Unbiased Quantitative Models of Protein Translation Derived from Ribosome Profiling Data. *PLoS Computational Biology*, 11(8):1–26, 2015. ISSN 15537358. doi: 10.1371/journal.pcbi.1004336.

[9] Nicholas T Ingolia, Sina Ghaemmaghami, John RS Newman, and Jonathan S Weissman. Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. *science*, 324(5924):218–223, 2009.

[10] Frank Spitzer. Interaction of markov processes. *Advances in Mathematics*, 5(2):246–290, 1970.

[11] Luca Ciandrini, Ian Stansfield, and M Carmen Romano. Ribosome traffic on mrnas maps to gene ontology: genome-wide quantification of translation initiation rates and polysome size regulation. *PLoS computational biology*, 9(1):e1002866, 2013.

[12] Aashiq H Kachroo, Jon M Laurent, Christopher M Yellman, Austin G Meyer, Claus O Wilke, and Edward M Marcotte. Systematic humanization of yeast genes reveals conserved functions and genetic modularity. *Science*, 348(6237):921–925, 2015.

[13] David Botstein. Why yeast?, 1991.

[14] Mark Johnston. Genome sequencing: The complete code for a eukaryotic cell. *Current Biology*, 6(5):500–503, 1996. ISSN 09609822. doi: 10.1016/S0960-9822(02)00526-2. URL http://linkinghub.elsevier.com/retrieve/pii/S0960982202005262.

[15] Ugrappa Nagalakshmi, Zhong Wang, Karl Waern, Chong Shou, Debasish Raha, Mark Gerstein, and Michael Snyder. The transcriptional landscape of the yeast genome defined by rna sequencing. *Science*, 320(5881):1344–1349, 2008.

[16] Moran Yassour, Tommy Kaplan, Hunter B Fraser, Joshua Z Levin, Jenna Pfiffner, Xian Adiconis, Gary Schroth, Shujun Luo, Irina Khrebtukova, Andreas Gnirke, et al. Ab initio construction of a eukaryotic transcriptome by massively parallel mrna sequencing. *Proceedings of the National Academy of Sciences*, 106(9):3264—3269, 2009.

[17] J Michael Cherry, Eurie L Hong, Craig Amundsen, Rama Balakrishnan, Gail Binkley, Esther T Chan, Karen R Christie, Maria C Costanzo, Selina S Dwight, Stacia R Engel, et al. Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic acids research*, page gkr1029, 2011.

[18] Ronny Lorenz, Stephan HF Bernhart, Christian Hoener Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, Ivo L Hofacker, et al. Viennarna package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011.

[19] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[20] Gene Ontology Consortium et al. Gene ontology consortium: going forward. *Nucleic acids research*, 43(D1):D1049–D1056, 2015.

[21] Naglis Malys. Shine-dalgarno sequence of bacteriophage t4: Gagg prevails in early genes. *Molecular Biology Reports*, 39(1):33–39, 2012. ISSN 1573-4978. doi: 10.1007/s11033-011-0707-4. URL http://dx.doi.org/10.1007/s11033-011-0707-4.

[22] Milisav. Cellular stress response. *Environmental Toxicology and ...*, (April), 1994. URL http://onlinelibrary.wiley.com/doi/10.1002/etc.5620130801/abstract.

[23] Xiangwei Gao, Ji Wan, Botao Liu, Ming Ma, Ben Shen, and Shu-Bing Qian. Quantitative profiling of initiating ribosomes in vivo. *Nature Methods*, 12(2):147–153, 2014. ISSN 1548-7091. doi: 10.1038/nmeth.3208. URL http://www.nature.com/doifinder/10.1038/nmeth.3208.

[24] Premal Shah, Yang Ding, Malwina Niemczyk, Grzegorz Kudla, and Joshua B Plotkin. Rate-limiting steps in yeast protein translation. *Cell*, 153(7):1589–1601, 2013.

[25] Karl Pearson. Notes on regression and inheritance in the case of two parents. In *Proceedings of the Royal Society of London*, volume 58, pages 240–242. Taylor & Francis, 1895.

[26] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. ISSN 00029556. URL http://www.jstor.org/stable/1412159.

[27] scikit-learn, sklearn.ensemble.randomforestregressor, . URL http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

[28] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[29] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. In C J C Burges, L Bottou, M Welling, Z Ghahramani, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 431–439. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/4928-understanding-variable-importances-in-forests-of-randomized-trees.pdf.

[30] I Borg and P Groenen. Modern multidimensional scaling. series in statistics, 1997.

[31] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates. In *Human-Machine Interactive Systems*, pages 199–233. Springer, 1991.

[32] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[33] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning. *Elements*, 1:337–387, 2009. ISSN 03436993. doi: 10.1007/b94608. URL http://statweb.stanford.edu/~tibs/ElemStatLearn.

[34] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[35] scikit-learn, sklearn.svm.svr, . URL http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html.

[36] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

[37] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, pages 2962–2970, 2015.

Chapter 6

Appendix A: Correlation and Feature Importance Plots

FIGURE 6.1: Correlation Coefficient and P-values for the Gritsenko et al. dataset [8]. With green color are represented the *Pearson's* r values and with red the *Spearman's* ρ ones.

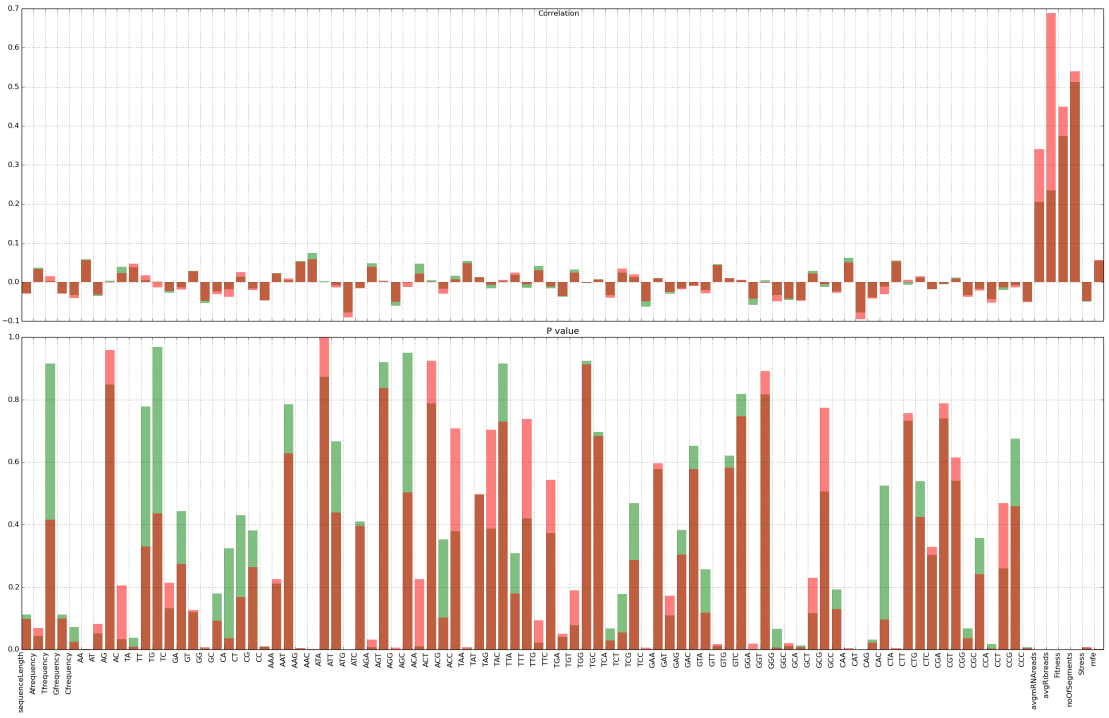


FIGURE 6.2: Correlation Coefficient and P-values for the Gritsenko et al. dataset [8]. With green color are represented the *Pearson's r* values and with red the *Spearman's ρ* ones. This includes the features *avgmRNAreads*,*avgRibreads*,*Fitness* and *noOfSegments*.

FIGURE 6.3: Correlation Coefficient and P-values for the Ciandrini et al. dataset [11]. With green color are represented the *Pearson's r* values and with red the *Spearman's ρ* ones.

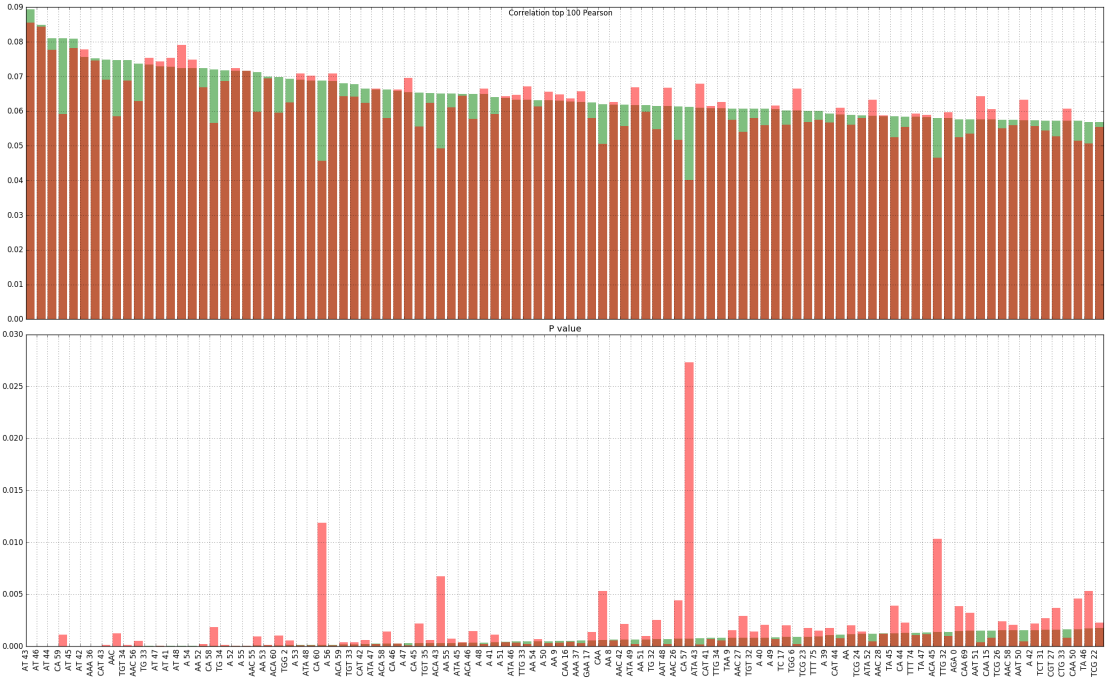


FIGURE 6.4: Correlation Coefficient and P-values for the Gritsenko et al. dataset [8] after we applied the sliding window method. Here we select the top best 100 features according to Person's values. With green color are represented the *Pearson's r* values and with red the *Spearman's ρ* ones.

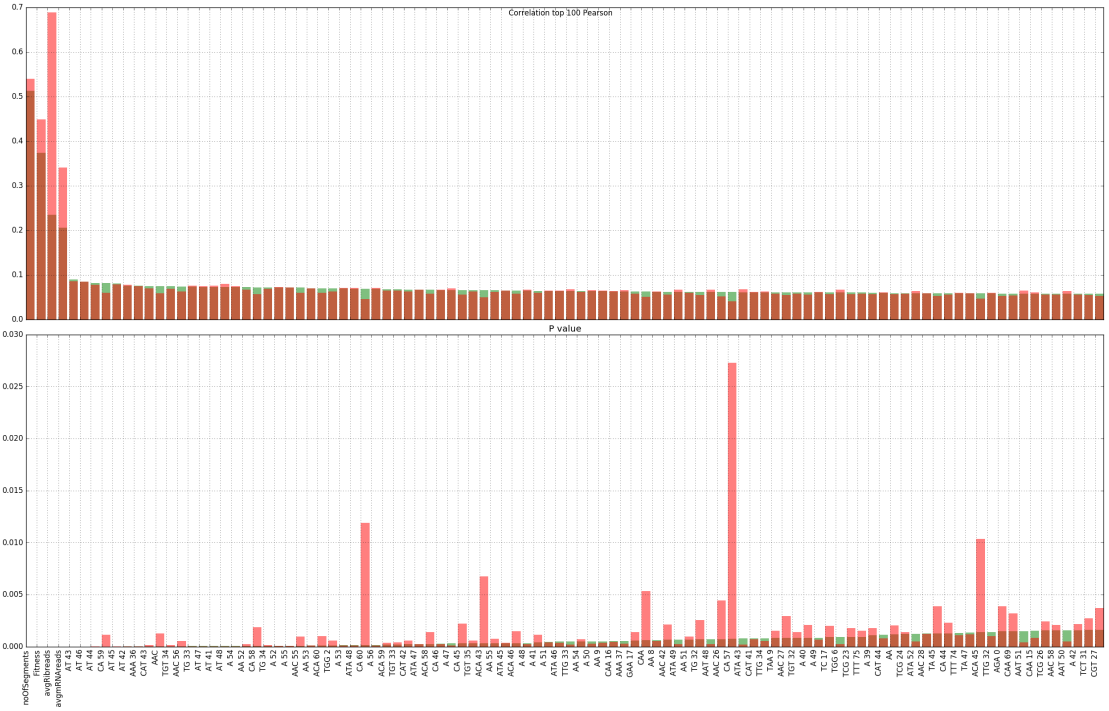


FIGURE 6.5: Correlation Coefficient and P-values for the Gritsenko et al. dataset [8] after we applied the sliding window method. Here we select the top best 100 features according to Person's values. With green color are represented the *Pearson's r* values and with red the *Spearman's ρ* ones.

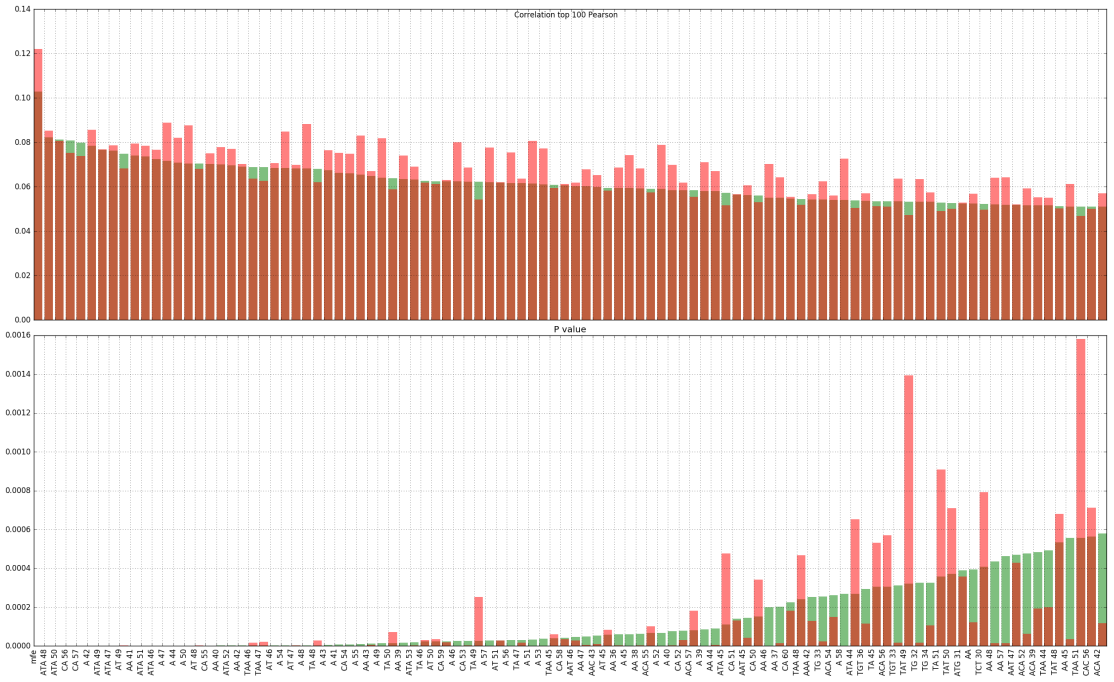


FIGURE 6.6: Correlation Coefficient and P-values for the Ciandrini et al. dataset [11] after we applied the sliding window method. With green color are represented the *Pearson's r* values and with red the *Spearman's ρ* ones.

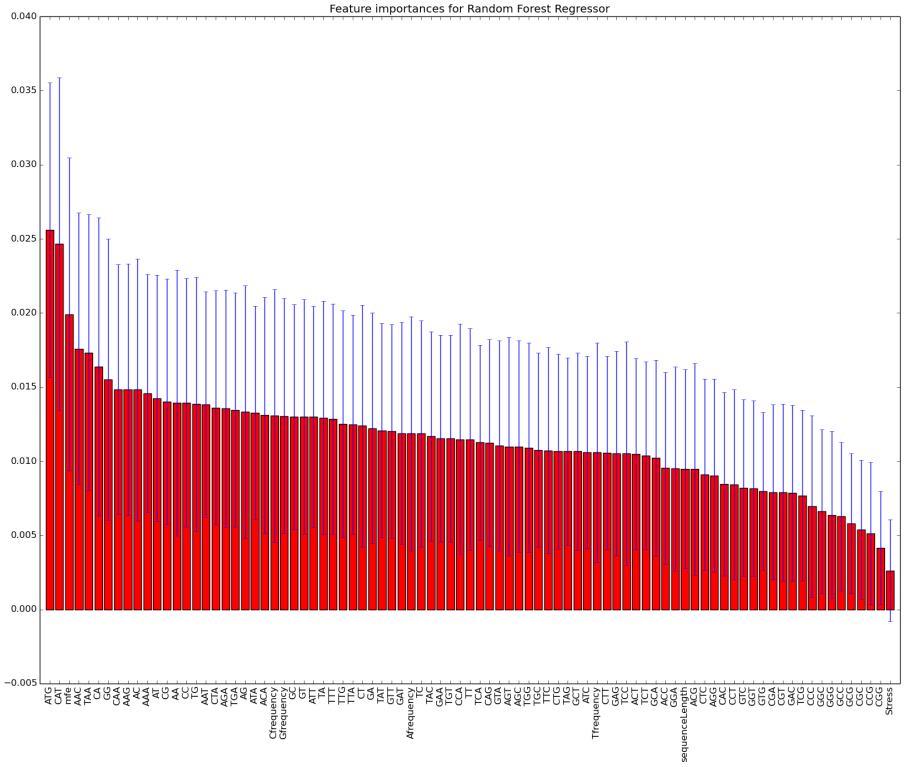


FIGURE 6.7: Average feature Importances and their average standard deviation as it occurs from the 5 fold Cross Validation, calculated by a Random Forest Regressor on the Gritsenko et al. dataset [8]. Here we select the top best 100 features according to Person's values.

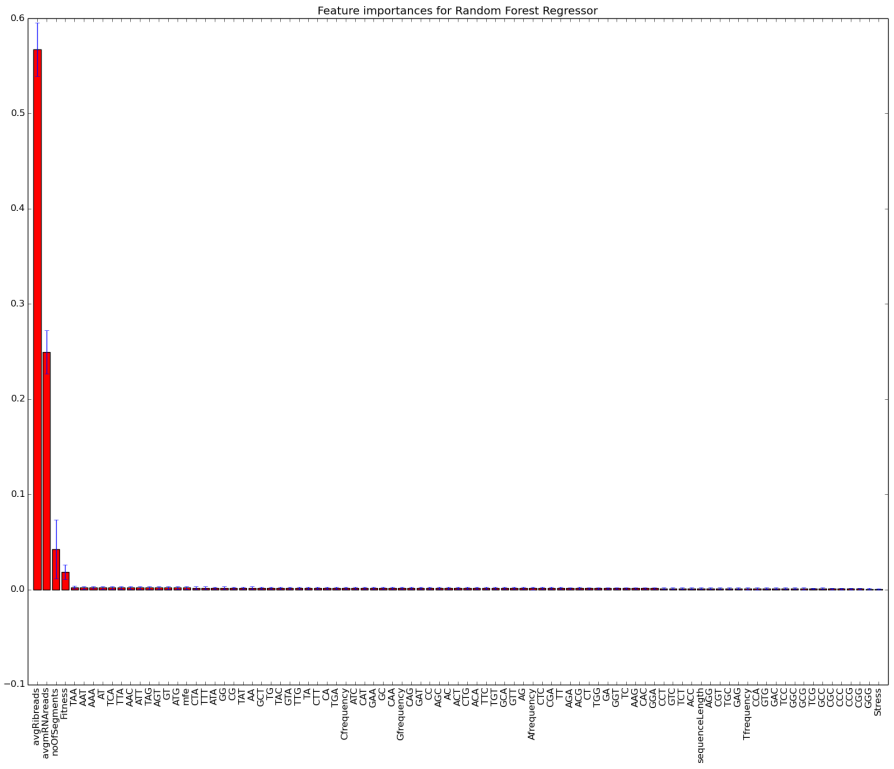


FIGURE 6.8: Average feature Importances and their average standard deviation as it occurs from the 5 fold Cross Validation, calculated by a Random Forest Regressor on the Gritsenko et al. dataset [8]. This time were included the four extra features: *Average mRNA Reads*, *Average Ribosome Reads*, *Fitness* and *Number of Segments*

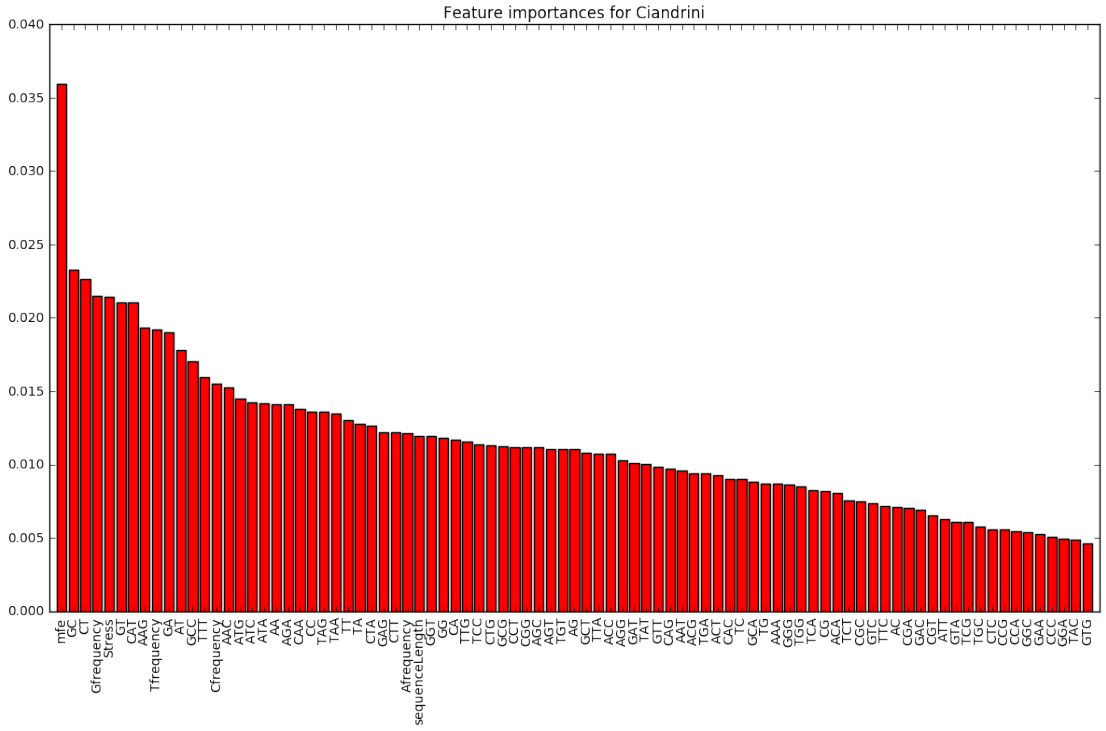


FIGURE 6.9: Average feature Importances and their average standard deviation as it occurs from the 5 fold Cross Validation, calculated by a Random Forest Regressor on the Ciandrini et al. dataset [11].

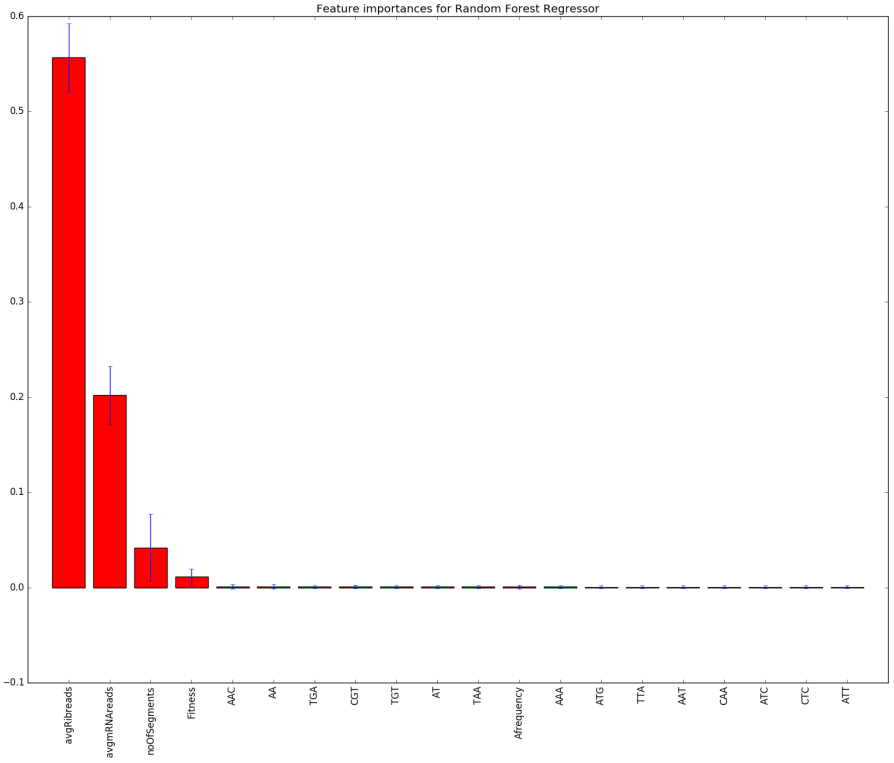


FIGURE 6.10: Average feature Importances and their average standard deviation as it occurs from the 5 fold Cross Validation, calculated by a Random Forest Regressor on the Gritsenko et al. dataset [8]. This time were included the four extra features: *Average mRNA Reads*, *Average Ribosome Reads*, *Fitness* and *Number of Segments*. For this calculation we have first applied sliding window which leads to many more features and thus we select only the top 20.

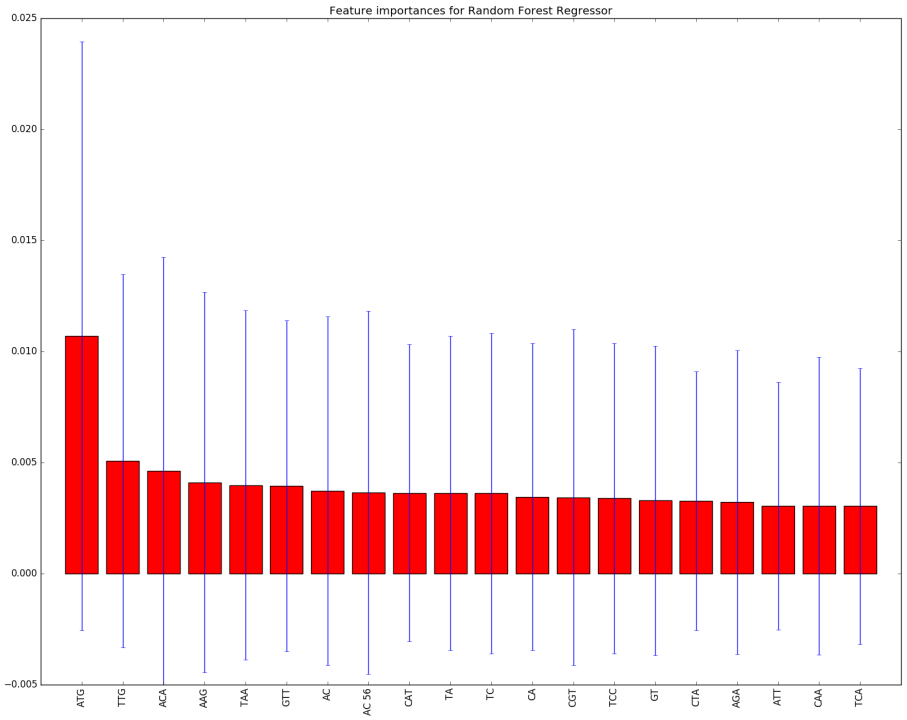


FIGURE 6.11: Average feature Importances and their average standard deviation as it occurs from the 5 fold Cross Validation, calculated by a Random Forest Regressor on the Gritsenko et al. dataset [8]. For this calculation we have first applied sliding window which leads to many more features and thus we select only the top 20.

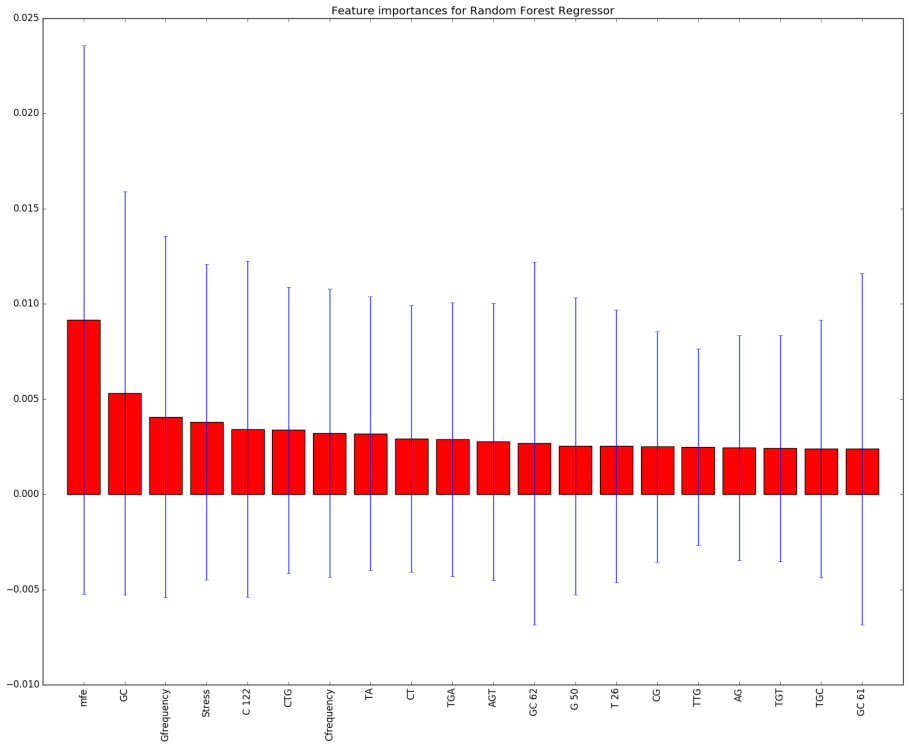


FIGURE 6.12: Average feature Importances and their average standard deviation as it occurs from the 5 fold Cross Validation, calculated by a Random Forest Regressor on the Ciandrini et al. dataset [11]. For this calculation we have first applied sliding window which leads to many more features and thus we select only the top 20.