# A comparison of classification methods for gene prediction in metagenomics

Fabiana Goés[1], Ronnie Alves[1,2], Leandro Corrêa[1], Cristian Chaparro[2] and Lucinéia Thom[3]

[1] PPGCC - Federal University of Pará, Belém, Brazil
fabii.goes@gmail.com
[2] Vale Institute of Technology, Belém, Brazil
ronnie.alves@itv.org, cristian.chaparro@itv.org
[3] PPGC - Federal University of Rio Grande do Sul, Porto Alegre, Brazil
lucineia@inf.ufrgs.br

**Abstract.** Metagenomics is an emerging field in which the power of genome analysis is applied to entire communities of microbes. It is focused on the understanding of the mixture of genes (genomes) in a community as whole. The gene prediction task is a well-known problem in genomics, and it remains an interesting computational challenge in metagenomics too. A large variety of classifiers has been developed for gene prediction though there is lack of an empirical evaluation regarding the core machine learning techniques implemented in these tools. In this work we present an empirical performance comparison of different classification strategies for gene prediction in metagenomic data. This comparison takes into account distinct supervised learning strategies: one lazy learner, two eager-learners and one ensemble learner. The ensemble-based strategy has achieved the overall best result and it is competitive with the prediction baselines of well-known metagenomics tools.

**Keywords**: Machine learning, classification methods, gene prediction, metagenomics

## 1 Introduction

Since the human genome project several computation strategies have been developed to shed a light on the amazing complexity of the complex human genome. Completed in 2003, this international research effort provided, for the fist time, the blueprint for building a human being. Nowadays, we are facing a new voyage of discovery into the microorganism world. Microbial communities support all life on Earth, and metagenomics is a revolutionary new approach to better understanding the microbial world. This new science opens doors to a large amount of scientific exploration and can help understand some of the most complex medical, agricultural, environmental, and economic challenges of today's world [1, 2].

Metagenomics is the application of shotgun sequencing to DNA obtained directly from an environmental sample or series of related samples, and it is also a

derivation of conventional microbial genomics, with the key difference being that it bypasses the requirement for obtaining pure cultures for sequencing [3]. It is focused on the understanding of the mixture of genes (genomes) in a community as a whole [4]. The gene prediction task is a well-known problem in genomics, and it remains an interesting computational challenge in metagenomics too.

Gene prediction is the procedure of finding protein and RNA coding sequences in the sample DNA. Depending on the applicability and success of the assembly, gene prediction can be done on post assembly *contigs*[1], on reads from unassembled metagenomes or on a mixture of *contigs* and individual unassembled reads. There are two main strategies for gene prediction [3]: i) **evidence-based** gene-calling methods use homology searches to find genes similar to those observed previously (reference microbial genomes); and ii) **ab initio** gene-calling relies on the intrinsic features of the DNA sequence to discriminate between coding and noncoding regions, allowing for the identification of homologs in the available databases. The former approach has two major drawbacks. Low values of similarity to known sequences either due to evolutionary distance or due to the short length of metagenomic coding sequences and the presence of sequence errors restrict the identification of homologs. In addition, novel genes without similarities are completely ignored. The latter approach usually employs Machine Learning (ML) algorithms which can smooth the previous gene prediction drawbacks. Still this requires a proper use of sophisticated classification methods and careful selection of potential DNA sequence features that could best discriminate between coding and noncoding sequences.

A large variety of classifiers has been developed for gene prediction. The hidden Markov models (HMM) is the state-of-the-art ML technique used since the 90's, and it is at the core of the pipeline called *GeneMark.hmm* [5]. Recently, metagenomic pipelines have adopted new classification strategies such as i) support vector machines(SVM) [6] ($MetaGUN$) and ii) artificial neural networks (ANN) [7](*Orphelia*). As an example, in *Orphelia* first a linear discrimination analysis takes place to select candidate features followed by ANN that calculates the probability of an ORF[2] being a potential coding sequence.

Applications of supervised machine learning methodologies continue to grow in the scientific literature across several domains. Jensen and Bateman conducted a careful text-mining over several biomedical articles in PubMed and they observed a moderate decrease in the use of both ANN and HMM, and an increase in usage of SVM and Random Forest in the literature [8]. This study takes into account, basically, the citation of these ML methods. In this work we present an empirical performance evaluation of the core ML techniques explored for gene prediction by some of the most used metagenomic pipelines.

[1] A contig is a continuous sequence resulting from the assembly of overlapping small DNA fragments (sequence reads).

[2] An ORF is a sequence of DNA that starts with a start codon, usually "ATG", and ends with any of the three termination codons (TAA, TAG, TGA).

## 2 Materials and Methods

In Figure 1 we depict the overall architecture devised for the comparison of the classifiers. It follows the classical steps of data preprocessing, learning and test. First, coding and non-coding sequences are extracted for the identification of potential sequence features, and next classification models are built for further prediction analysis (Figure 1-A). Once new sequences are retrieved it is possible to classify them in accordance with the classification models, and thus, an appreciation regarding whether it is a coding sequence or not can be done(Figure 1-B).

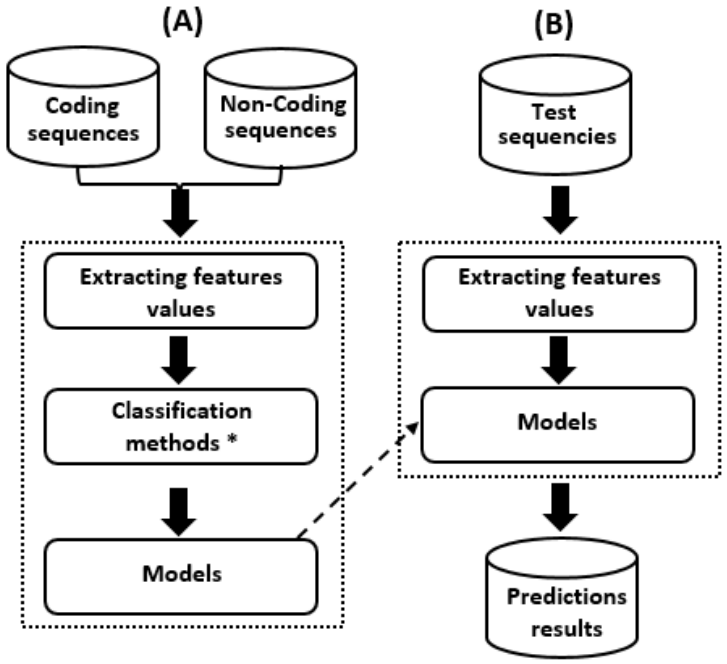


Fig. 1: The overall architecture devised for the comparison of the classification methods.

### 2.1 Classification methods

We have selected four classification strategies for the comparison study. These methods employ distinct learning strategies, and ideally, each one has a particular manner to generalize the search space. The gene prediction problem is simply a binary classification or *concept learning* (positive class: coding sequence and

negative class: no coding sequence). This comparison takes into account distinct supervised learning strategies: one lazy learner (KNN: K-Nearest Neighbors), two eager-learner (SVM: Support Vector Machines and ANN: Artificial Neural Networks) and one ensemble learner (RF: Random Forest).

**Random forest (RF)** It is a well-known ensemble approach for classification tasks proposed by Breiman [9]. Its basis comes from the combination of tree-structured classifiers with the randomness and robustness provided by bagging and random feature selection. Several decision trees are trained with random bootstrap samples from the original data set ( 2/3 of data) and afterwards, results are combined into a single prediction: for classification tasks, by means of voting; for regression tasks, by averaging the results of all trees. The fact that the predicted class represents the mode of all the classes output by individual trees gives robustness to this ensemble classifier in relation to a single tree classifier. Given its basis on an ensemble learning it is less impacted by $overfitting$, making it a potential candidate ML approach in bioinformatics problems [10]. Though, as far as we know it has not been explored as a solution in the gene prediction of metagenomic sequences.

**K-Nearest Neighbors (KNN)** Nearest-neighbor classifiers are based on learning by analogy, by comparing a given test instance with training instances that are similar to it [11]. The training instances are described by $n$ features. Each instance represents a point in a $n$-dimensional space. Thus, all of the training instances are stored in an $n$-dimensional pattern space. When an unknown instance is provided, a $k$-nearest-neighbor classifier searches the pattern space for the $k$ training instances closest to the unknown instance. $Closeness$ is usually defined in terms of a distance metric, such as Euclidean distance. $K$-nearest neighbors are also used as a baseline strategy for comparison among distinct classifiers.

**Artificial Neural Networks (ANN)** A neural network is a set of connected input/output units in which each connection has a weight associated with it. During the learning stage, the network learns by adjusting the weights with aims to predict the correct class label of the input instances. ANN also involves long training times and are also criticized for their poor interpretability. Nevertheless, it has a higher tolerance to noisy data as well as the ability to classify patterns on which it has not been trained. *Backpropagation* is the most popular ANN algorithm and it performs learning on a *multilayer feed-forward* neural network [11]. A *multilayer feed-forward* neural network basically consists of an input layer, one or more hidden layers, and an output layer.

**Support Vector Machines (SVM)** It uses a linear model to implement non-linear class boundaries. SVM transform the input using a nonlinear mapping, thus, turning the instance space into a new space. A linear model (the *maximum*

*margin hyperplane*) constructed in the new space can represent a nonlinear decision boundary in the original space. The *maximum margin hyperplane* is the one that gives the greatest separation between classes. The instances that are closest to this hyperplane, so the ones with minimum distance to it, are called *support vectors*. Other kernel functions can be used instead to implement distinct nonlinear mappings. Two that are often suggested are the *radial basis functions (RBF) kernel* and the *sigmoid kernel*. These functions do not present large differences in terms of prediction accuracy, though this observation depends on the application domain. SVM has been used extensively in several domains, and in some cases it outperforms ANN [12].

### 2.2 Feature engineering

Feature engineering is at the core of classification strategies and it is a crucial step on prediction modeling. Essentially, two different types of information are currently used to try to find genes in a genomic sequence: i) *extrinsinc content sensors* explore a sufficient similarity between a genomic sequence region and a protein or DNA sequence present in a database in order to determine whether the region is transcribed and/or coding; and ii) *intrinsic content sensors* proposed particularly for prokaryotic genomes, in which features that characterize the sequence as "coding" for a protein are carefully calculated for discrimination analysis [13, 14]. Examples of content sensors are: nucleotide composition and especially $(G + C)$ content (introns being more A/T-rich than exons, especially in plants), codon composition, hexamer frequency, base occurrence periodicity, etc. Hexamer usage has been widely exploited by a large number of algorithms through different methods [14]. Table 1 presents six content sensors that are strongly used by gene prediction tools in metagenomics. For the comparison study we focused on three main types of features: $(G + C)$ content, length and codon usage. From this information we derived a total of six features as follows: 1) GC content, 2) GC content in the first position of each codon, 3) GC content in the second position of each codon, 4) GC content in the third position of each codon, 5) the sequence length, 6) the codon usage variance among the 61 monocodons.

| | GC Content | Length | Codon usage | Dicodon usage | TIS | Aminoacid usage |
|---|---|---|---|---|---|---|
| *Orphelia* | x | x | x | x | x | |
| *MetaGUN* | | x | x | | x | |
| *MGC* | x | x | x | x | x | x |
| *MetaGene* | x | | x | x | | |
| *FragGeneScan* | | | x | | | |

**Table 1.** Content sensors features used [x] by gene prediction tools in metagenomics.

**GC-content** It is the percentage of guanine and cytosine bases in all bases of a sequence. It has been used extensively by several gene prediction tools. This utilization is mainly due to the fact that coding regions present, on average, a higher GC content than on non coding sequences [15]. Differently from previous studies (see Table 1), we calculated the total level of GC content, and the content at the first, second and third monocodon positions with the aim to evaluate their impact in the gene prediction task. In this way, four features are derived from the GC content.

**Length** Another feature for discrimination between coding and non-coding sequence is its length. The intergenic regions are usually smaller than coding regions[14].

**Codon Usage** Perhaps the most important features for the discrimination between coding and non-coding sequences can be calculated from codon usage [16], in particular the frequencies of $4^3$ monocodons. These frequencies represent the occurrences of successive trinucleotides (non-overlapping). For the characterization of monocodon usage, we compute the variance among the 61 monocodons, since gene sequences do not contain stop codons.

### 2.3 Training Data

The training data is basically DNA sequences having both coding sequences (positive) and intergenic regions (negative) instances. Our approach to compare the four classification methods is based on a learning scheme over eight prokaryotic genomes, namely two *Archaeas* and six *Bacterias*, available in GenBank[3] (Table 2). The choice of these organisms has to do with the experimental metagenomic data that will be evaluated while testing the predictive models. Thus, either these organisms belong to the same branch of the evolutionary tree or they are associated to Acid Mine Drainage biofilms (Section 2.4).

We have developed an algorithm to properly extract the coding and non-coding regions, on both forward and reverse strands, from these eight "complete" genomes. This algorithm was applied to regions with sequence lengths higher than 59 bp. Sequences less than 60 bp are ignored since they are too short to provide useful information [6]. Those originating from the annotated genes are used as positive instances of coding sequences, whereas others are treated as items of the non-coding class. After running this procedure we came up with a total of 30144 sequences, being 10106 related to intergenic regions and remaining 20038 of coding sequences.

### 2.4 Test Data

The metagenomic data selected for the comparison study is the Acid Mine Drainage (AMD) biofilm [17], freely available at the site of NCBI [4]. This biofilm

[3] http://www.ncbi.nlm.nih.gov/news/10-22-2013-genbank-release198
[4] http://www.ncbi.nlm.nih.gov/books/NBK6860/

| Species | GenBank Acc. |
|---|---|
| Thermoplasma acidophilum * | NC_002578 |
| Thermoplasma volcanium * | NC_002689 |
| Acidimicrobium ferrooxidans | NC_013124 |
| Acidithiobacillus caldus | NC_015850 |
| Acidithiobacillus ferrooxidans | NC_011206 |
| Acidithiobacillus ferrivorans | NC_015942 |
| Candidatus Nitrospira defluvii | NC_014355 |
| Thermodesulfovibrio yellowstonii | NC_011296 |

**Table 2.** The prokaryotic genomes used as reference for the training data. The "*" symbol highlights the two *Archaeas*.

sequencing project was designed to explore the distribution and diversity of metabolic pathways in acidophilic biofilms. Acidophilic biofilms are self-sustaining communities that grow in the deep subsurface and receive no significant inputs of fixed carbon or nitrogen from external sources. While some AMD is caused by the oxidization of rocks rich in sulfide minerals, this is a very slow process and most AMD is due directly to microbial activity [18]. More information regarding the AMD study as well as environmental sequences, metadata and analysis can be obtained at [17].

We have selected prokaryotic genomes associated to the same species found in Tyson[17]. Thus, five genomes (2 *Archaeas* and 3 *Bacterias*) were extracted from GenBank to create the test data (Table 3).

| Species | GenBank Acc. |
|---|---|
| FA: Ferroplasma acidarmanus * | NC_021592 |
| TA: Thermoplasmatales archaeon BRNA * | NC_020892 |
| LFI: Leptospirillum ferriphilum | NC_018649 |
| LFO: Leptospirillum ferrooxidans | NC_017094 |
| SA: Sulfobacillus acidophilus | NC_015757 |

**Table 3.** The prokaryotic genomes used as reference for the test data. The "*" symbol highlight the two *Archaeas*.

### 2.5 Measures of prediction performance

The classifiers will be evaluated through the evaluation of classical prediction performance measures, namely, accuracy (ACC), specificity (SPE), sensitivity (SEN) and Kappa. All these measures are easily calculated from the resulting confusion matrix for each classifier. This matrix usually has two rows and two columns that reports the number of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN). Though we provide the results for all these measures, we believe that Kappa is the most suitable measure to compare distinct classifiers. Kappa measures how closely the instances labeled by the classifiers matched the data labeled as *ground truth*, controling for the

ACC of a random classifier as measured by the expected accuracy. Thus, the kappa for one classifier is properly comparable to others kappa's classifiers for the same classification task.

$$ACC = \frac{TP+TN}{P+N} \quad (1)$$

$$SPE = \frac{TN}{TN+FP} \quad (2)$$

$$SEN = \frac{TP}{TP+FN} \quad (3)$$

$$Kappa = \frac{Pr(a)-Pr(e)}{1-Pr(e)} \quad (4)$$

## 3 Results and Discussion

### 3.1 Performance of the classifiers

The prediction modeling and evaluation was carried out with the *caret R* package [19]. We use the built-in *tune*() function for resampling and tuning to optimize all classifiers parameters. The best values were as follows: i) RF (mtry 4), KNN (k=5), ANN (size=5 and decay=0.1), SVML (C=0.5). The performance measures were calculated from the average performance of three repetition of a 10-fold cross validation scheme (Table 4). The most promising results were obtained by the RF model using 200 trees (Figure 2).

| | ACC | KAPPA |
|---|---|---|
| RF model | **0.94** | **0.87** |
| KNN model | 0.87 | 0.70 |
| ANN model | 0.91 | 0.80 |
| SVML model | 0.88 | 0.74 |

**Table 4.** The average performance of the classifiers. The orange cells highlight the best performance achieved by the RF classifier.

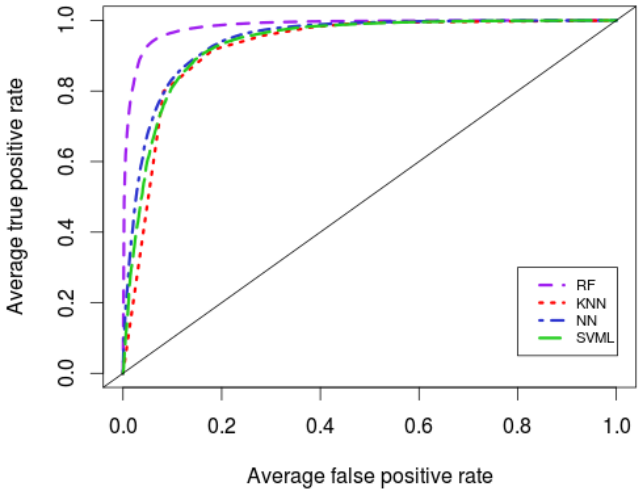


Fig. 2: RF has the best performance among all classifiers.

### 3.2 Comparison of classifiers using independent test data

In this section we present the performance comparison of the selected classifiers using the independent test data discussed in Section 2.4. So, the main goal was to evaluate how classifiers correctly classify the known coding sequences for the species associated to the AMD metagenome. As we expected the ensemble learning classifier employed by RF has achieved the best performance among all classifiers (Table 5 and Table 6). Ensembles are designed to increase the accuracy of a single classifier by training several distinct classifiers and combine their decisions to output a single class label. Given such accuracy-oriented design, ensemble learning algorithms are less likely to $overfitting$ when dealing with imbalanced data.

The SVM has an overall performance similar to KNN (base classifier), and this is partially due to the generalization carried by a linear SVM. Probably a radial SVM model would be able to generalize better the search space. On the other hand, the other eager learner, ANN, presents competitive results. As an example, ANN outperforms RF for the LFI specie (Kappa=0.9097).

Let us assume that we built a gene prediction method that is solely based on a RF model, so the overall performance of our model would have SEN=0.91 for a "hypothetical" metagenome (as the one discussed in Section 2.4). Table 7 presents a prediction baseline discussed in [6], where the $MetaGUN$ tool, based on a SVM classifier, outperforms the other two well-known gene prediction *pipelines*. Though SVM would hypothetically outperform our RF model, it is

important to mention that our feature set is less complex than the one employed by $MetaGUN$.

| | SEN | | | | SPE | | | |
|---|---|---|---|---|---|---|---|---|
| Species | RF | ANN | KNN | SVML | RF | ANN | KNN | SVML |
| FA | **0.9380** | 0.8801 | 0.6337 | 0.6521 | 0.9047 | 0.8642 | 0.9503 | 0.8662 |
| LFI | **0.8945** | 0.8783 | 0.8139 | 0.8203 | 0.9304 | 0.9316 | 0.9352 | 0.9276 |
| LFO | **0.8797** | 0.8469 | 0.7939 | 0.7743 | 0.9599 | 0.9628 | 0.9570 | 0.9504 |
| SA | **0.9317** | 0.9017 | 0.8145 | 0.8517 | 0.9433 | 0.9398 | 0.9486 | 0.9267 |
| TA | **0.9085** | 0.8408 | 0.7711 | 0.7642 | 0.9889 | 0.9679 | 0.9640 | 0.9522 |

**Table 5.** The comparison performance of classifiers in accordance to the SEN and SPE measures. The highlighted cells show the best results.

| | ACC | | | | Kappa | | | |
|---|---|---|---|---|---|---|---|---|
| Species | RF | ANN | KNN | SVML | RF | ANN | KNN | SVML |
| FA | 0.9173 | 0.8702 | 0.8302 | 0.785 | **0.8275** | 0.7298 | 0.6182 | 0.5317 |
| LFI | 0.9156 | 0.9097 | 0.8854 | 0.8835 | **0.8256** | **0.9097** | 0.7599 | 0.7565 |
| LFO | 0.9263 | 0.9143 | 0.8888 | 0.8767 | **0.8472** | 0.8213 | 0.7666 | 0.741 |
| SA | 0.9383 | 0.9235 | 0.8913 | 0.8947 | **0.8741** | 0.8434 | 0.7746 | 0.7834 |
| TA | 0.957 | 0.9175 | 0.8875 | 0.9175 | **0.9089** | 0.8243 | 0.7577 | 0.737 |

**Table 6.** The comparison performance of classifiers in accordance to the ACC and Kappa measures. The highlighted cells show the best results.

| | 1200 bp | | 870 bp | | 535 bp | | 120 bp | |
|---|---|---|---|---|---|---|---|---|
| | Sen | Spec | Sen | Spec | Sen | Spec | Sen | Spec |
| MetaGUN (SVM) | **97.7** | 94,8 | 97.4 | 95.2 | 96.9 | 95.4 | 93.2 | 89.6 |
| FragGeneScan (Markov Models) | **95.7** | 87.3 | 95.5 | 88.0 | 95.2 | 88.4 | 90.4 | 82.1 |
| Orphelia (Neural Network) | **94.6** | 94.7 | 94.1 | 94.7 | 93.3 | 94.6 | 82.0 | 76.4 |

**Table 7.** The performance of three well-known pipelines for gene prediction in metagenomic data. The highlighted cells show the best results obtained for detecting the real signal (gene).

## 4 Conclusions

Gene prediction is a well-known computational challenge in both genome and metagenome analysis. This latter poses an even more difficult problem since: i) metagenomes are a mixture of several distinct genomes and ii) most of the available genomes are not completed, so mainly $draft$ genomes are available. Therefore, the task of gene prediction is *biased* in the proper selection of potential

features in this complex domain as well as the choice of a robust machine learning algorithm.

In this work we presented an empirical comparison of several well-known classification methods applied to gene discovery in experimental metagenomic data. Though the performance of the four base classifiers was good, the ensemble-based strategy *Random Forest* has achieved the overall best result. As it can be observed by the associated ML literature. Ensemble learning strategies such as Random forest has been successfully applied on a large variety of business and scientific applications. Basically such observation is due to the fact the combination of models could be able to best generalize the hypothesis space. Though the best approach to combine models continues an open problem in ensemble learning research.

We plan to develop a new gene prediction *pipeline* having its basis on Random Forest. To the extent of our knowledge there is no reference of gene prediction algorithms based on a RF classifier.

## Author's contributions

FG and RA performed the analysis and developped the pipeline. RA and CC supervised the study. FG, RA, CC and LT wrote the manuscript.

## Acknowledgements

This work is partially supported by the Brazilian National Research Council (CNPq – *Universal calls*) under the BIOFLOWS project [475620/2012-7]. FG has a master scholarship by the Federal Agency for Support and Evaluation of Graduate Education (Capes).

## References

1. Handelsman, J., Tiedje, J., Alvarez-Cohen, L., Ashburner, M., Cann, I., Delong, E., Doolittle, W., Fraser-Liggett, C., Godzik, A., Gordon, J., et al.: The new science of metagenomics: Revealing the secrets of our microbial planet. Nat Res Council Report **13** (2007)
2. Thomas, T., Gilbert, J., Meyer, F.: Metagenomics-a guide from sampling to data analysis. Microb Inform Exp **2**(3) (2012)
3. Kunin, V., Copeland, A., Lapidus, A., Mavromatis, K., Hugenholtz, P.: A bioinformatician's guide to metagenomics. Microbiology and Molecular Biology Reviews **72**(4) (2008) 557–578
4. Wooley, J.C., Godzik, A., Friedberg, I.: A primer on metagenomics. PLoS computational biology **6**(2) (2010) e1000667
5. Lukashin, A.V., Borodovsky, M.: Genemark. hmm: new solutions for gene finding. Nucleic acids research **26**(4) (1998) 1107–1115
6. Liu, Y., Guo, J., Hu, G., Zhu, H.: Gene prediction in metagenomic fragments based on the svm algorithm. BMC bioinformatics **14**(Suppl 5) (2013) S12

7. Hoff, K.J., Lingner, T., Meinicke, P., Tech, M.: Orphelia: predicting genes in metagenomic sequencing reads. Nucleic acids research **37**(suppl 2) (2009) W101–W105
8. Jensen, L.J., Bateman, A.: The rise and fall of supervised machine learning techniques. Bioinformatics **27**(24) (2011) 3331–3332
9. Breiman, L.: Random forests. Machine Learning **45**(1) (2001) 5–32
10. Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. BMC bioinformatics **9**(1) (2008) 307
11. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. Morgan kaufmann (2012)
12. Faceli, K.: Inteligência artificial: uma abordagem de aprendizado de máquina. Grupo Gen-LTC (2011)
13. Ermolaeva, M.D., White, O., Salzberg, S.L.: Prediction of operons in microbial genomes. Nucleic acids research **29**(5) (2001) 1216–1221
14. Mathé, C., Sagot, M.F., Schiex, T., Rouzé, P.: Current methods of gene prediction, their strengths and weaknesses. Nucleic acids research **30**(19) (2002) 4103–4117
15. Fickett, J.W.: Recognition of protein coding regions in dna sequences. Nucleic acids research **10**(17) (1982) 5303–5318
16. Hoff, K.J., Tech, M., Lingner, T., Daniel, R., Morgenstern, B., Meinicke, P.: Gene prediction in metagenomic fragments: a large scale machine learning approach. BMC bioinformatics **9**(1) (2008) 217
17. Tyson, G.W., Chapman, J., Hugenholtz, P., Allen, E.E., Ram, R.J., Richardson, P.M., Solovyev, V.V., Rubin, E.M., Rokhsar, D.S., Banfield, J.F.: Community structure and metabolism through reconstruction of microbial genomes from the environment. Nature **428**(6978) (2004) 37–43
18. Johnson, D.B., Hallberg, K.B.: Acid mine drainage remediation options: a review. Science of the total environment **338**(1) (2005) 3–14
19. Kuhn, M.: The caret package homepage. URL http://caret. r-forge. r-project. org (2010)