Simplifying p-value calculation for the unbiased microRNA enrichment analysis, using ML-techniques

Konstantinos Zagganas
University of the Peloponnese & "Athena" RC
kzagganas@uop.gr

Maria Lioli
National Kapodistrian University of Athens
mlioli@di.uoa.gr

Thanasis Vergoulis
"Athena" RC
vergoulis@athenarc.gr

Theodore Dalamagas
"Athena" RC
dalamag@athenarc.gr

ABSTRACT

The investigation of the role of small bio-molecules (called microRNAs) in biological functions is a very popular topic in bioinformatics, since microRNAs have been shown to present novel therapeutic methods for diseases like cancer or Hepatitis C. In order to predict the involvement of microRNAs in biological functions many statistical approaches have been used that involve p-value calculations, with the most popular one being Fisher's exact test. However, it has been shown that data distribution does not match with any of the theoretical distributions used by the aforementioned approaches. Thus, an empirical randomization approach is preferred. Nevertheless, such analyses are computationally intensive. In this paper, we present a novel approach for microRNA enrichment analysis using Machine Learning techniques, in order to predict p-values instead of calculating them using randomization experiments. This simplifies the work for bioinformatics data analysts, helping them to efficiently perform multiple enrichment analysis tasks.

1 INTRODUCTION

Transcriptomics, a popular sub-field of bioinformatics, deals with RNA biomolecules that are produced by DNA transcription and either contain instructions on how to produce proteins or perform other functions vital to life. One such class of RNAs are microRNAs (or miRNAs) which are short biomolecules that "target" specific genes and block their expression. This means that they affect biological functions by preventing protein production. Dysregulation of certain miRNAs has been linked to diseases like cancer, neuronal diseases, inflammatory diseases, etc [2]. Consequently, predicting how groups of miRNAs affect certain biological functions is of great importance to researchers. It has led to the development of prediction workflows that combine data about miRNA gene-targets, as well as associations between genes and biological functions, and in order to measure the association between a group of miRNAs and a specific biological function using statistical tests.

The most popular statistical test used in the relevant literature is Fisher's exact test [3]. The test uses the hypergeometric distribution to produce p-values that indicate the strength of the association between the group of miRNAs and a biological function.The hypergeometric distribution assumes that every gene has an equal probability of being targeted by a miRNA. However, complex biological mechanisms and effects (modeled by prediction algorithms) make such an assumption unrealistic.

Indeed, it was shown in 2015 [1] that the data related to miRNA targets contain a bias in how genes are targeted, and this means that the hypergeometric test leads to biased or invalid results. However, the authors of [1] proposed an unbiased, empirical miRNA functional enrichment approach, which uses a permutation (randomization) test to produce *empirical p-values*.

Generally, such approaches measure the strength of association based on the real data, without making assumptions about the structure of the data like theoretical statistical tests. In order to test whether a group of miRNAs is associated to a specific biological process, the method calculates a statistical measure relevant to the miRNA group and compares it to the statistical measure calculated for a large number of randomly assembled miRNA groups. Thus, if the number of random groups presenting a better behaviour (in terms of the statistical measure) is small, then the significance of the association between the query and the biological function under examination is large.

However, this approach is computationally intensive, because it involves a very large number of operations between sets (union and intersection) leading to large execution times (e.g., in the order of hours, on 8 cores of a Intel Xeon CPU). For this reason, attempts have been made to alleviate this issue, either by making the set operations faster [5] or by using indexing techniques to reduce the number of redundant operations performed [7]. These attempts led to a reduction in execution times by an order of magnitude, but still a significant amount of time is required to complete such a task (e.g., several minutes are required on a single Xeon CPU core). Given that researchers usually need to run multiple analyses, the total computational cost can still be very high.

In this paper, we present a novel approach for miRNA enrichment analysis using machine learning techniques, to predict p-values using features of miRNA groups, relevant to the problem, instead of calculating them using randomization experiments. This simplifies the work for bioinformatics data analysts, helping them to efficiently perform multiple enrichment analysis tasks. Our contributions are: (a) framing the problem, (b) data set creation and feature engineering, (c) determining a shortlist of promising machine learning models, using cross-validation, and (d) fine-tuning to determine the best models for our case. Our approach shows that the best model demonstrates an R^2 score above 90%, and $MAE = 0.048$.

2 BACKGROUND

2.1 Permutation test

Given a miRNA group as a query, the biological function permutation test introduced in [1] consists of the following steps:

(1) Calculate a statistical measure S (e.g., left-sided overlap - see below), that captures a type of 'relevance' of the biological function with the query, according to the genes that are related to both of them.
(2) Create a large number (e.g., 1 million) of randomly assembled miRNA groups, with each containing the same number of miRNAs, and calculate S for each of these groups, as well.
(3) Measure the proportion of randomly assembled groups that present more favourable values for S than the query.

More formally, each miRNA is represented as the set of genes targeted by it. Consequently, a group of miRNAs is represented as a set, containing the union of all genes targeted by each miRNA in the group. Moreover, a biological function is also represented as the set of genes participating in that function.

Based on the previous, we can now describe one popular permutation test: Let M be the set of genes containing the union of targets from all miRNAs in a query group, and also let B the set of genes participating in a biological function. The statistical measure used to compare the query to the randomly assembled miRNA groups is the *left-sided overlap* and it is defined as follows:

$$left\text{-}sided\ overlap(M,B) = \frac{sizeof(M \cap B)}{sizeof(M)}$$

Essentially, the left-sided overlap is defined as the proportion of targeted genes that also participate in the biological function. Then, we create 1 million random miRNA groups M_j with the same size as the query and calculate the left-sided overlap for each of them. The empirical p-value is defined as (where overlap is the left-sided overlap):

$$p - value = \frac{sizeof(\{M_j : overlap(M_j,B) \geq overlap(M,B)\})}{n}$$

which is the proportion of randomly assembled groups presenting a larger left-sided overlap than the query.

2.2 Performance issues

The above analysis relies on a very large number of union and intersection set operations. Given that this analysis is performed for more than one biological functions, and that more than 20K biological functions exist, a few million union and about 20 billion intersection operations are performed in the span of the analysis.

The software implemented by the authors in [1] is written in Python, uses hash join set operations and a typical analysis on a single CPU core of an Intel Xeon CPU requires many hours. However, based on the fact that this analysis is very repetitive, even a small increase in operation speed is going to lead to a large total speedup. With this motivation, in [5] we re-implemented the algorithm in C++. We also improved the analysis performance by exploiting bit vectors, as well as a hybrid version of hash join between sets of items and bit vectors. This made the analysis one order of magnitude faster requiring about 40 minutes to complete on a single core of the same Xeon computer.

Then in [7], we introduced novel indexing techniques, that allowed us to remove a large number of redundant operations performed by our previous version and thus managed to reduce the time to approximately half of what was required previously on a single core. Furthermore, we used a technique that allowed us to run the analysis on only a subset of the biological functions (which are predicted to be statistically significant) and managed to further reduce the time required for p-value calculation to about 3 minutes (on a single core of the Xeon processor). The downside to the latter approach, is that p-values are produced only for the functions expected to be significant and not all biological functions in the dataset. Consequently, we were motivated to use ML to train a model that will predict p-values very quickly and for every biological function in the data set.

3 MACHINE LEARNING FOR P-VALUE PREDICTION

3.1 Features

In order to train a machine learning model to predict p-values (`p_value`) as accurately as possible, we selected features from our dataset based on their biological meaning and their relevance to the analysis. The features are summarized below:

miRNA group size (`mirna_group_size`): The number of miRNAs in a miRNA group.

Biological function ID (`biological_process`): a unique string that identifies each biological function. This string consists of 2 letters (same for all biological functions) and a numerical part. We turned this ID into a numerical value, by stripping the letters from the string.

Number of common genes (`number_of_common_genes`): The number of genes targeted by a miRNA group that also belong to the biological function.

Left-sided overlap (`left_sided_overlap`): the left-sided overlap as defined in Section 2.1.

Right-sided overlap (`right_sided_overlap`): Given M and B from Section 2.1, the right-sided overlap is defined as:

$$right\text{-}sided\ overlap(M,B) = \frac{sizeof(M \cap B)}{sizeof(B)}$$

Two-sided overlap (`overlap`): Given M and B from Section 2.1, the two-sided overlap (or Jaccard coefficient) is defined as:

$$two\text{-}sided\ overlap(M,B) = \frac{sizeof(M \cap B)}{sizeof(M \cup B)}$$

Common genes as a percentage of the universe of genes (`common_genes_proportion_to_total`): The number of common genes between M and G as the proportion of the total number of genes in the universe of genes.

Common gene list (`common_genes`): The list of common genes, sorted by alphabetical order. We used label encoding to turn the values into categorical values.

Chromosomes of common genes (`common_chr`): The list of chromosomes on which the common genes are located, sorted by alphabetical order. We used label encoding to turn the values into categorical values.

Number of chromosomes where common genes are located (`number_of_common_chr`): The number of chromosomes on which the common genes are located.

Left chromosome overlap (`chr_left_sided_overlap`) : The number of chromosomes on which the common genes are located divided by the number of chromosomes of the genes targeted by the miRNA group.

Right chromosome overlap (`chr_right_sided_overlap`): The number of chromosomes on which the common genes are located, divided by the number of chromosomes of the genes belonging in the biological function.

Two sided chromosome overlap (`chr_overlap`): The number of chromosomes on which the common genes are located, divided by the number of chromosomes for the union of the genes contained in M and B.

Using these features, we produced a dataset for groups of miRNAs of size 10, 25, 50, 100 (containing 100 groups of each size) for all biological functions as described in the Gene Ontology [6]. Finally, to handle categorical features (miRNA group size, biological function, common gene list, chromosomes of common genes) we used label encoding.

3.2 ML Algorithms

In order to find the best method to use for our case, we are going to use the following algorithms:

- **Linear/Ridge/Lasso Regressors:** traditional regression methods, fitting a linear equation that uses the least squares method (with several variants of regularization).
- **Decision Tree Regressor:** uses decision trees to make predictions.
- **Random Forest Regressor:** estimates target value by combining average estimation values of several individual prediction models based on classifying decision trees for a number of subsets of the data set.
- **Adaboost Regressor:** combines multiple weak decision trees into one.
- **Gradient Boost (XGBoost, LightGBM, CatBoost):** in XGBoost, the estimation of the target is done by combining estimates of many individual prediction models based on decision trees. LightGBM is similar to XGBoost, but prediction is much faster than XGBoost. Regarding CatBoost, the key difference is that it builds decision symmetric trees. Both LightGBM and CatBoost can inherently handle categorical features.
- **MLPerceptron Regressor:** typical Neural Net configuration.

4 EVALUATION

In this section, we present preliminary results on a dataset that has been created by only using 1000 random miRNA groups to calculate p-values. This was done due to time constraints and to reach some preliminary conclusions about the dataset, in order to compare the algorithms presented in Section 3.2. Given the definition of an empirical p-value in Section 2.1, we expect that the p-values in the dataset will present a small accuracy, since the number of random miRNA groups is three orders of magnitude smaller than the required number. This is expected to create larger errors than the ones that the permutation test produces. However, these preliminary results are a first indicator on which algorithms present the worst performance in terms of accuracy and thus, which algorithms will be featured in the final work.

4.1 Linear correlation

In Figure 1 we can see a heat map of the linear correlation between the features. Each cell represents the correlation measured by Pearson correlation coefficient r between features in row x and column y. The values of r range between -1 and 1. The larger the deviation from 0 the stronger the positive (for values closer to 1) or negative (for values closer to -1) correlation is. It is interesting to note here that the p-value, which is the feature we want to predict, does not seem to have a strong linear correlation (either positive or negative) with any of the features individually. Nevertheless, this does not mean that the p-value is not related with the features in a non-linear way. More specifically, it is expected to be related with the left-sided overlap (see p-value definition in Section 2.1).

4.2 Preliminary results

In this section we are going to perform a preliminary evaluation of the algorithms we outlined in Section 3.2. In order to evaluate and compare the algorithms, we are going to use the following statistical measures [4]:

(1) Mean Absolute Error (*MAE*): the *MAE* is the mean absolute difference between observed and predicted values. Essentially, it is used to compare predictions to actual observed values.
(2) Coefficient of determination (R^2): the R^2 score is a statistical measure of how close are the predictions to the real data points. It is essentially a goodness-of-fit measurement.

Model selection and implementation was performed with the method of k-Fold cross-validation. We first split the dataset into the *training* and *testing* dataset. The training dataset is split into k folds (groups). Then k-1 of those groups are used to train the model and the group left is used for *validation*. The score of the validation is recorded. Then, the process is repeated, but this time another group is used for validation and the rest for retraining, until all folds have been exhausted. The average of all validation scores is the final validation score. Finally, the *test score* is retrieved by testing the model against the testing data set. Final validation scores for each algorithm are shown in Table 1 and the testing scores in Table 2. Model parameters were selected via Grid Search for the Linear Regressors and the rest via Random Search, due to the large number of parameters and memory constraints.

It should be noted that the algorithms based on linear regression models do not perform as well as the other algorithms. Our approach shows that the best model is LightGBM, demonstrating a $MAE = 0.048$.

5 CONCLUSION & FUTURE WORK

In this paper, we presented an approach for miRNA enrichment analysis using machine learning techniques, in order to predict p-values instead of calculating them using randomization experiments. The goal is to simplify the work for bioinformatics data analysts, facilitating multiple enrichment analysis. Preliminary results showed that the best model demonstrates an R^2 score above 90%, and $MAE = 0.048$. As next steps, we plan to expand our dataset to include p-values estimated using 1 million random groups in order increase their accuracy as well as the accuracy of the prediction.

ACKNOWLEDGMENTS

We acknowledge support of this work by the project “Moving from Big Data Management to Data Science" (MIS 5002437/3) which is implemented under the Action“Reinforcement of the Research and Innovation Infrastructure”, funded by the Operational Programme “Competitiveness, Entrepreneurship and Innovation” (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

REFERENCES

[1] Thomas Bleazard, Janine A Lamb, and Sam Griffiths-Jones. 2015. Bias in microRNA functional enrichment analysis. *Bioinformatics* 31, 10 (01 2015).
[2] Shailendra Dwivedi, Purvi Purohit, and Praveen Sharma. 2019. MicroRNAs and Diseases: Promising Biomarkers for Diagnosis and Therapeutics. *Indian Journal of Clinical Biochemistry* 34, 3 (01 Jul 2019), 243–245.
[3] R. A. Fisher. 1992. *Statistical Methods for Research Workers.* Springer New York, New York, NY, 66–70.

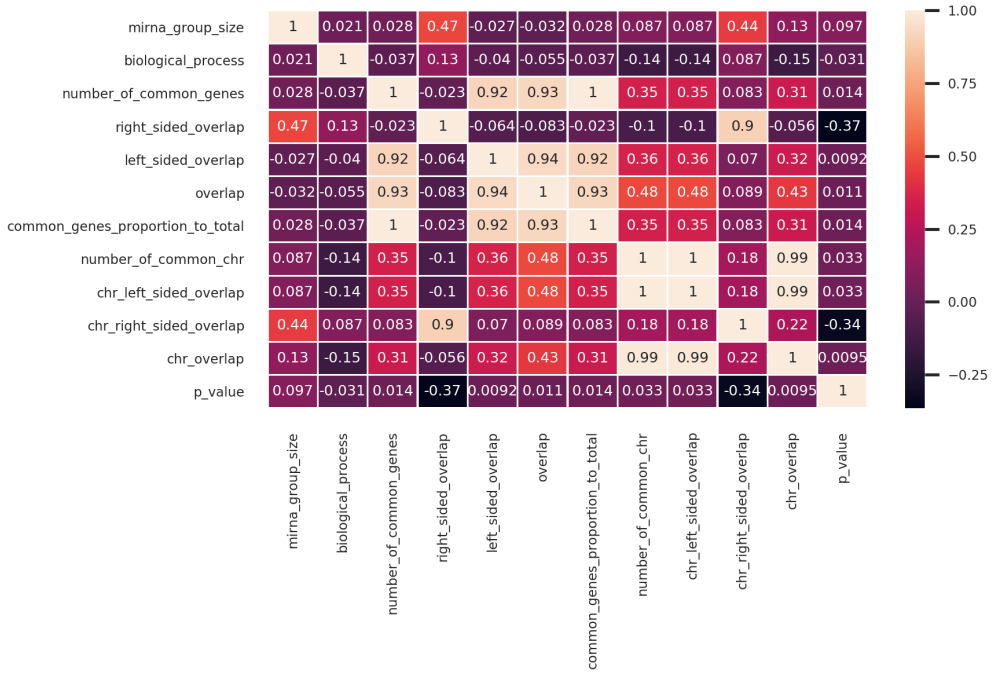

Figure 1: Linear correlation between the selected features

	MAE	R^2	Model parameters
Linear Regressor	0.242	0.246	copy_x = True, fit_intercept = False, normalize = True
Ridge Regressor	0.242	0.2462	alpha=1, copy_x = True, fit_intercept = False, normalize = True, solver = auto, tol = 0.001
Lasso Regressor	0.4364	$4 \times e^{-6}$	default
Decision Tree Regressor	0.1027	N/A	min_samples_leaf = 2, min_samples_split = 5,max_depth = None,max_features = None
Random Forest Regressor	0.0992	N/A	n_estimators = 50, n_jobs = -1, min_samples_leaf = 2, min_samples_split = 2, verbose = 10
Adaboost Regressor	0.13	N/A	n_estimators = 20, loss = square, base_estimator = RandomForestRegressor, (n_jobs =-1, verbose=2, max_depth=25, n_estimators=4)
LightGBM Regressor	0.038	N/A	objective = poisson, boosting_type = gbdt, learning_rate = 0.5, n_estimators = 900, n_jobs = -1, num_leaves = 40, max_depth=20, reg_alpha = 1, reg_lambda = 1, force_col_wise = True
XGBoost Regressor	0.0745	N/A	verbosity = 2, max_depth = 20, n_estimators=200
CatBoost Regressor	0.13	N/A	n_estimators=1000, learning_rate=0.5, bootstrap_type=Bernoulli,
ML Perceptron Regressor	0.1817	N/A	hidden_layer_sizes = (24,24,24), activation = "relu", max_iter = 40, learning_rate='adaptive'

Table 1: Algorithm evaluation: validation set

	MAE	R^2	Model parameters
Random Forest Regressor	0.1443	N/A	n_estimators = 50, n_jobs = -1, min_samples_leaf = 2, min_samples_split = 2, verbose = 10
LightGBM Regressor	0.0477	N/A	objective = poisson, boosting_type = gbdt, learning_rate = 0.5, n_estimators = 900, n_jobs = -1, num_leaves = 40, max_depth=20, reg_alpha = 1, reg_lambda = 1, force_col_wise = True
XGBoost Regressor	0.136	N/A	objective = poisson, boosting_type = gbdt, learning_rate = 0.5, n_estimators = 900, n_jobs = -1, num_leaves = 40, max_depth=20, reg_alpha = 1, reg_lambda = 1, force_col_wise = True

Table 2: Algorithm evaluation: test set

[4] A. Géron. 2019. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, Incorporated.

[5] K.Zagganas, T. Vergoulis, M. D. Paraskevopoulou, I. S. Vlachos, S. Skiadopoulos, and T. Dalamagas. 2017. BUFET: boosting the unbiased miRNA functional enrichment analysis using bitsets. *BMC Bioinformatics* (2017).

[6] The Gene Ontology Consortium. 2018. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research* 47, D1 (11 2018), D330–D338.

[7] K. Zagganas, T. Vergoulis, S. Skiadopoulos, and T. Dalamagas. 2020. Efficient Calculation of Empirical P-Values for Association Testing of Binary Classifications. In *32nd International Conference on Scientific and Statistical Database Management (SSDBM 2020).* Association for Computing Machinery (ACM).