SOFTWARE TOOL ARTICLE

Meta-Signer: Metagenomic Signature Identifier based on rank aggregation of features [version 1; peer review: 1 approved with reservations, 1 not approved]

Derek Reiman ¹, Ahmed Metwally², Jun Sun³, Yang Dai ¹

¹Department of Bioengineering, University of Illinois at Chicago, Chicago, Illinois, 60612, USA

²Department of Genetics, Stanford University School of Medicine, Stanford, California, 94305, USA

³Department of Medicine, University of Illinois at Chicago, Chicago, Illinois, 60612, USA

V1 First published: 09 Mar 2021, 10:194
<https://doi.org/10.12688/f1000research.27384.1>

Latest published: 09 Mar 2021, 10:194
<https://doi.org/10.12688/f1000research.27384.1>

Abstract

The advance of metagenomic studies provides the opportunity to identify microbial taxa that are associated with human diseases. Multiple methods exist for the association analysis. However, the results could be inconsistent, presenting challenges in interpreting the host-microbiome interactions. To address this issue, we develop Meta-Signer, a novel Metagenomic Signature Identifier tool based on rank aggregation of features identified from multiple machine learning models including Random Forest, Support Vector Machines, Logistic Regression, and Multi-Layer Perceptron Neural Networks. Meta-Signer generates ranked taxa lists by training individual machine learning models over multiple training partitions and aggregates the ranked lists into a single list by an optimization procedure to represent the most informative and robust microbial features. A User will receive speedy assessment on the predictive performance of each machine learning model using different numbers of the ranked features and determine the final models to be used for evaluation on external datasets. Meta-Signer is user-friendly and customizable, allowing users to explore their datasets quickly and efficiently.

Keywords

Metagenome-wide Association Study, Feature Extraction, Machine Learning, Rank Aggregation

This article is included in the **Bioinformatics** gateway.

Open Peer Review

Approval Status

	1	2
version 1		
09 Mar 2021	view	view

1. **Braden T Tierney** , Harvard Medical School, Boston, USA
2. **Boyang Ji** , Chalmers University of Technology, Gothenburg, Sweden

Any reports and responses or comments on the article can be found at the end of the article.

This article is included in the **Artificial Intelligence and Machine Learning** gateway.

Corresponding author: Yang Dai (yangdai@uic.edu)

Author roles: **Reiman D:** Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Metwally A:** Conceptualization, Writing – Review & Editing; **Sun J:** Conceptualization, Writing – Review & Editing; **Dai Y:** Conceptualization, Investigation, Project Administration, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2021 Reiman D *et al.* This is an open access article distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Reiman D, Metwally A, Sun J and Dai Y. **Meta-Signer: Metagenomic Signature Identifier based on rank aggregation of features [version 1; peer review: 1 approved with reservations, 1 not approved]** F1000Research 2021, 10:194 <https://doi.org/10.12688/f1000research.27384.1>

First published: 09 Mar 2021, 10:194 <https://doi.org/10.12688/f1000research.27384.1>

Introduction

Recent metagenomic studies of the gut microbiome have linked dysbiosis to many human diseases¹⁻³. The identification of microbial taxa associated with human disease has been one of the important efforts in metagenomics data analysis⁴. Various metagenomic studies use parametric or non-parametric statistical tests to detect differentially abundant individual taxa between disease and control groups⁵⁻⁹. These types of methods can potentially miss taxa with weak associations but together can present strong statistical association.

In order to capture group association, several methods are proposed by exploring related taxa on a phylogenetic taxonomic tree¹⁰⁻¹⁴.

These elaborated statistical methods enhanced the detection of the microbial group association. However, they may still fail to detect complex multivariate nonlinear associations.

Alternative approaches of using machine learning (ML) models have been advocated for the prediction of the host phenotype¹⁵. This is motivated by the findings that a microbial signature for the host phenotype may be complex, involving simultaneous over- and under-representations of multiple microbial taxa potentially interacting with each other. Classical ML models, such as Random Forest (RF), Logistic Regression and Support Vector Machines (SVMs), and deep neural networks (DNNs) have been applied to host phenotype prediction using microbial abundance features¹⁶⁻²².

While the ML approaches demonstrated promising results on host phenotype prediction^{23,24}, it is a challenging task for users to determine what is the best ML model and how many features are needed in order to achieve robust prediction, especially on external validation datasets. In addition, each ML algorithm may

generate different feature importance rankings^{12,22,25}, complicating the decision on a consistent and informative signature for the host phenotype of interest.

In this work, we introduce a novel tool, Meta-Signer, a Metagenomic Signature Identifier based on rank aggregation of informative taxa learned from individual ML models. Meta-Signer uses RF, SVM, penalized Logistic Regression, and multiple-layer perceptron neural network (MLPNN) models to evaluate importance of each microbial taxon and generates a ranked list of features (i.e., taxa) per model. It aggregates all the ranked lists using a procedure “RankAggreg” based on the cross-entropy method or the genetic algorithm²⁶. Finally, Meta-Signer reports the top ranking features specified by the user and generates the ML models using these features. Meta-Signer’s workflow is shown in **Figure 1**.

Meta-Signer is user-friendly and easy to run. It provides a readable summaries as HTML outputs as well as final trained ML models. Meta-Signer is distributed as Python tool and available at <https://github.com/YDaiLab/Meta-Signer>.

Methods

Implementation

The inputs to Meta-Signer are (1) a tab separated file of taxa abundance values where each row represents a taxon and each column represents a sample, and (2) a line separated list of response values where each row represents the phenotypic response of a sample. The first column in the abundance table should be the taxonomic identification of the taxon. An additional required file is (3) the run configuration file with user specified parameters. A final optional file is (4) the model parameters for the neural network architectures in JSON format. If this file is not found, Meta-Signer will tune the parameters and save them for later use.

Figure 1. The Meta-Signer workflow. Large rounded rectangles represent different modules of the workflow. Microbial abundance is preprocessed and filtered, and then used to train ML models. Features are ranked for each model and an overall aggregated feature ranking is constructed. Meta-Signer generates portable, user-friendly HTML files for visualization as well as ML models trained on a subset of high ranking features. SVM, support Vector Machine; MLPNN, multiple-layer perceptron neural network; ML, machine learning.

Meta-Signer includes three classic ML models (RF, Linear SVM, Logistic Regression), as well as an MLPNN model. RFs are decision tree learning models trained in an ensemble fashion, taking the average of the ensemble to give a robust decision tree²⁷. While growing each tree, a decision is made at each node by selecting the feature from a random subset of features that best splits the data into two subsets based on the Gini impurity of each subset. Given a set of data points with k classes, let p_i be the proportion of samples of class i for $i \in \{1...k\}$. The Gini impurity of the set is calculated as

$$I_G(p) = 1 - \sum_{i=1}^k p_i^2 \quad (1)$$

Our method implements the RF model using the *scikit-learn* Python library. Once trained, features are then extracted by evaluating the mean decrease impurity. For each node, the importance of the feature node being split on the decision tree is calculated as the decrease in Gini impurity before and after the split. This value is then weighted by the proportion of total samples that were split upon that node. A feature's importance is then calculated by averaging the weighted importance values of nodes that split using that feature across all trees in the ensemble.

SVMs are supervised ML models that learn the best hyper-plane to separate two classes of data²⁸. In case of linear SVMs, a set of weights (w), and an intercept (b) will be learned. The class of the sample x_i can then be determined as

$$\hat{y} = \text{sign}(w^T x_i + b) \quad (2)$$

Since the weights can be used to rank the importance of features, we used the linear SVMs in Meta-Signer for feature extraction. Once trained, features can be ranked using the absolute value of the learned weight parameters. SVM models are tuned with a grid search using the *scikit-learn* Python library.

Logistic Regression fits a logistic function to estimate the probability of binary classification; however, it can be extended to multi-class scenarios²⁹. The model predicts the probability of a sample x_i being class 1 as

$$\hat{y} = \frac{1}{1 + e^{-\beta x + b}} \quad (3)$$

where β are the weight parameters which are learned. We apply shrinkage to reduce the total number of model parameters in the final model using L1 regularization in order to penalize the absolute value of the weights, eliminating a portion of the features to create a sparse model. Given a set of samples x_i ($i = 1, ..., n$) where each sample has m features and a binary class label y_i , the model minimizes the cost

$$C = \frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] + \lambda \sum_j |\beta_j| \quad (4)$$

where the weight parameters are penalized with the regularization parameter λ . Logistic regression models are trained using

the *scikit-learn* Python library. Once trained, the β values are used to rank features based on their absolute value. Neural networks are consisted of multiple layers of nodes that are fully connected with edges constituting weights³⁰. The values of a hidden layer are a linear combination of the values from the previous layer which is passed through a non-linear activation function. More explicitly, the values of a hidden layer h_l is calculated as

$$h_l = \psi(W_l^T h_{l-1} + b_l) \quad (5)$$

where h_{l-1} are the values from the previous hidden layer, W_l are the weights connecting h_{l-1} to h_l , b_l is a bias value, and ψ is a non-linear activation function. Meta-Signer uses the Rectified Linear Unit activation function for hidden layers and the softmax activation function on the output layer. The entire network is trained using a single loss function

$$C = -\ln(a_c) + \lambda \sum_{l \in L} \|W_l\|_2 \quad (6)$$

Here a_c is the predicted softmax probability of a sample's true class c . The second term performs L2 regularization on the network weights and is penalized by λ . Networks are trained using early stopping with a validation subset of 10% of the training data.

Before training, Meta-Signer checks for a file containing network hyper-parameters for MLPNN models. If it does not find this file, Meta-Signer will use the first partition of the cross-validation to empirically determine the hyper-parameters. This is done using another cross-validation on the training set of the first partition. In addition, using the configuration file, the user can set custom parameters and even disable any of the learning models if they do not wish to incorporate it into their results. In Meta-Signer, RF, SVM, and Logistic Regression are trained using the *scikit-learn* python package and MLPNN models are trained using *Tensorflow*.

For each ML model in each cross-validated partition, Meta-Signer extracts the feature scores and uses the scores to construct a ranked feature list. RF features were scored using a method called mean decrease impurity. For each node, the importance of the feature being split upon is calculated as the decrease in Gini impurity from before and after the split. This value is then weighted by the proportion of total samples that were split upon that node. A feature's importance is then calculated by averaging the weighted importance values of nodes that split using that feature across all trees in the ensemble. Features in Logistic Regression and SVM models were scored based on the magnitude of their coefficients in the decision functions.

The extraction of features from DNN models is a challenging task in general. We use a procedure developed in 31 to evaluate features in MLPNNs. Briefly, the MLPNN features were evaluated by calculating the cumulative weight across all layers by taking the running product of all the weight matrices in the

learned networks. The product results in a matrix that has a column for each class and a row for each feature, and the value at a given index is that feature's cumulative impact for that class. We then consider a feature's importance as the maximum impact across classes to create a single ranked list.

For each partition of the cross-validation, we generate a single ranked list for each of ML models. Once the entirety of the cross-validated training is complete, the entire set of all ranked lists across all models is aggregated into a single top- k ranked list by minimizing the distance between the set of ranked lists and the top- k list, where k is specified by the user in the configuration file. More specifically, given a set of ranked lists $\{\ell_1, \dots, \ell_m\}$, the top- k ranked list, $\hat{\theta}$, is determined as,

$$\hat{\theta} = \arg \min_{\theta \in L} \sum_{i=1}^m w_i d(\theta, \ell_i) \quad (7)$$

Here, L is the state space of top- k rankings, w_i is a weight associated with ℓ_i , and $d(\theta, \ell_i)$ is the distance between a proposed top- k ranked list, θ , and ℓ_i .

The aggregation is performed using the R package RankAggreg²⁶. This package uses either a genetic algorithm or cross-entropy based approach with Markov Chain Monte Carlo sampling to find the top- k features that minimize the sum of the distances between each of the input sets and the generated top- k set. The distance used is the Spearman's Correlation. Each input ranked list is weighted in the aggregation by the area under the receiver operating curve (AUC). After the model predictions are evaluated and the features are ranked into a single list, Meta-Signer provides a summary of the results in a portable HTML file. The file contains a description of the run and evaluation metrics for the different models in the form of boxplots. It also provides the distribution of the feature importance scores for each ML model. Lastly, it provides a list of the top- k taxa selected from the original taxa, the proportion of individual ranking sets that each taxon was present in the top- k , the rank and p-value under a PERMANOVA test³², and the class in which the taxon was found to be predictive for. All images are encoded into the file, allowing the HTML file to be moved without considering the location of the images.

Operation

Meta-Signer's general workflow proceeds as described above and as shown in Figure 1. Meta-Signer was designed for Python 3.7 and requires the following Python packages: *numpy*, *pandas*, *scipy*, *scikit-learn*, *scikit-bio*, *Tensorflow*, *matplotlib*, *seaborn*. In addition, a working version of R version 3.0 or higher is required. Code and instructions for configuration and running can be found at <https://github.com/YDaiLab/Meta-Signer>.

Use cases

In this section we will demonstrate how to run Meta-Signer on a dataset of patients with inflammatory bowel disease (IBD) which is provided in the GitHub repository. This dataset came from the Prospective Registry in IBD Study at MGH (PRISM)³³, which enrolled patients with a diagnosis of either Crohn's

disease (CD) or ulcerative colitis (UC). The dataset includes 68 samples with CD, 53 samples with UC, and 34 healthy samples.

In addition, Meta-Signer includes another IBD dataset for external testing. This dataset consists of two independent cohorts from the Netherlands³⁴. The first cohort consists of 22 healthy subjects who participated in the general population study LifeLines-DEEP in the northern Netherlands. The second cohort consists of subjects with IBD from the Department of Gastroenterology and Hepatology, University Medical Center Groningen, Netherlands and includes 20 samples with CD and 23 samples with UC. Together, both the PRISM dataset and the external IBD dataset included 201 microbial features. Datasets were evaluated using all three classes as well as in a binary case by combining CD and UC samples.

To use Meta-Signer and the data provided, the user needs to download Meta-Signer from the GitHub repository using the following command:

```
git clone https://github.com/YDaiLab/Meta-Signer.git
cd Meta-Signer
```

The user must make sure that all the Python and R dependencies are met before running Meta-Signer (see Operation). The user can either install these manually, or use the provided Meta-Signer Conda environment file provided using the following command:

```
conda env create -f meta-signer.yml
source activate meta-signer
```

Before running Meta-Signer, the user should open the file *config.py* in the *src* directory. This file can be used to change the run parameters of Meta-Signer and turn off and on different ML methods. An example is shown in Figure 2.

Using these parameters, we will run 10 iterations of 10-fold cross-validation on the data found in the *PRISM_3* directory under the data folder. Any taxon not found in at least 10% of samples will be removed as well as any taxon with less than 0.001 mean abundance. We will use the genetic algorithm method for rank aggregation to generate a candidate list with a maximum of 50 features. Descriptions for each parameter can

```
NumberTestSplits = 10
NumberRuns = 10
Normalization = Standard
DataSet = PRISM_3
FilterThreshCount = 0.1
FilterThreshMean = 0.001
MaxK = 50
AggregateMethod = GA
```

Figure 2. User specified configuration parameters.

be found on Meta-Signer's GitHub page as well as in the configuration file. The user can then generate the aggregated ranked list using the following command:

```
python generate_feature_ranking.py
```

This will perform the cross-validation evaluation and feature ranking, storing all the results in the *results* folder under a directory named after the dataset. This directory will contain

internal cross-validation results as well as an HTML file that allows the user to see how different ML methods perform on the dataset. Performance of the 10 iterations of 10-fold cross-validation on the PRISM dataset is shown in [Table 1](#). We include an analysis of both binary class (IBD vs Healthy) and tertiary class (IBD vs CD vs UC) scenarios.

The HTML output displays these results in the form of boxplots. An example for the PRISM dataset is shown in [Figure 3](#).

Table 1. Mean cross-validated results over the PRISM dataset. Standard deviation is shown in parentheses. RF, Random Forest; SVM, Support Vector Machine; MLPNN, multiple-layer perceptron neural network; AUC, area under the receiver operating curve; MCC, Matthews correlation coefficient; F1, F1 score.

		RF	SVM	LogisticRegression	MLPNN
PRISM	AUC	0.91 (0.08)	0.81 (0.13)	0.82 (0.11)	0.87 (0.10)
	MCC	0.50 (0.28)	0.29 (0.32)	0.28 (0.27)	0.39 (0.30)
	Precision	0.84 (0.10)	0.76 (0.13)	0.76 (0.11)	0.80 (0.12)
	Recall	0.85 (0.07)	0.81 (0.08)	0.78 (0.07)	0.82 (0.07)
	F1	0.83 (0.09)	0.77 (0.10)	0.76 (0.08)	0.80 (0.09)
PRISM (3 Class)	AUC	0.88 (0.06)	0.67 (0.09)	0.72 (0.10)	0.74 (0.10)
	MCC	0.55 (0.17)	0.19 (0.19)	0.30 (0.19)	0.35 (0.19)
	Precision	0.72 (0.11)	0.47 (0.15)	0.56 (0.14)	0.60 (0.13)
	Recall	0.70 (0.11)	0.49 (0.11)	0.55 (0.12)	0.58 (0.12)
	F1	0.69 (0.11)	0.46 (0.12)	0.53 (0.12)	0.57 (0.12)

Figure 3. HTML output of machine learning cross-validated evaluation. Buttons on the top allow users to cycle through different metrics. RF, Random Forest; SVM, Support Vector Machine; MLPNN, multiple-layer perceptron neural network; AUC, area under the receiver operating curve; MCC, Matthews correlation coefficient; F1, F1 score.

In addition, Meta-Signer will evaluate the different ML methods' training performance using increasing numbers of ranked features. Since these models will eventually overfit, the saturation of performance can help guide the user on how many features are needed to train the final model on. For the PRISM dataset, we see this saturation around 30 features, as shown in [Figure 4](#)

Using the point of saturation on the AUC curve as a guide, the user can then select the number of features and train final models using the following command:

```
python generate_final_models.py PRISM_3 30
```

Here *PRISM_3* points to the result directory labeled *PRISM_3* that Meta-Signer has just generated and 30 represents the number of features the user would like to have in the final models. In addition, the user can provide an external dataset to evaluate,

```
python generate_final_models.py
PRISM_3 30 -e PRISM_3_external
```

Note that *PRISM_3_external* must be a directory in the *data* directory with its own *abundance.tsv* and *labels.txt* files. This command will generate a directory that will contain the final trained ML models as well as another HTML file showing the ranked lists for each model as well as the aggregated ranked list as shown in [Figure 5](#).

In addition, if an external dataset is available for evaluation, the HTML file will display a table of metrics for both

evaluation on the training set and evaluation on the external dataset. An example is shown in [Figure 6](#).

Discussion

We have developed Meta-Signer, a user-friendly tool for the extraction of robust microbial taxa that are predictive to host phenotype from multiple ML models. By training different types of ML models, Meta-Signer exploits the similarities in the ranked lists of taxa learned by individual ML models to create a single aggregated set of informative microbial taxa for host phenotype prediction. We have shown that Meta-Signer can provide more informative features when compared to similar methods in both binary and multi-class scenarios.

To evaluate Meta-Signer, we benchmark against other a previously published methods Biosigner³⁵ and a non-parametric PERMANOVA test³². Biosigner is a generic ML driven feature selection method for omics data and available in R. It uses trained RF, SVM and Partial-Least Squared Discriminant Analysis models to selectively eliminate features, resulting in a single set of remaining features. The non-parametric PERMANOVA test was included as a baseline method of feature ranking for comparison. Results for the benchmarking can be found in the *Extended data* document on the Meta-Signer GitHub page.

Conclusions

In conclusion, Meta-Signer is a user-friendly tool to identify a robust set of highly informative microbial taxa that are predictive of human disease status from a metagenomic dataset.

Figure 4. HTML output of training performance to help the user select their desired number of ranked features. RF, Random Forest; SVM, Support Vector Machine; MLPNN, multiple-layer perceptron neural network; AUC, area under the receiver operating curve.

Aggregated Feature List

Microbe	% in top-k	Elevated class	PERMANOVA rank	Adjusted p-value
<i>Eubacterium rectale</i>	100.0	Control	24	7.000e-03
<i>Dialister invisus</i>	94.2	CD	39	2.900e-02
<i>Bifidobacterium dentium</i>	73.0	Control	53	6.900e-02
<i>Roseburia hominis</i>	77.8	Control	16	1.000e-03
<i>Subdoligranulum unclassified</i>	64.5	Control	11	1.000e-03
<i>Faecalibacterium prausnitzii</i>	76.5	UC	20	3.000e-03
<i>Eubacterium siraeum</i>	65.8	UC	86	2.230e-01
<i>Coprococcus catus</i>	85.5	Control	9	1.000e-03
<i>Acidaminococcus unclassified</i>	38.5	CD	44	4.200e-02
<i>Ruminococcus torques</i>	34.0	Control	76	1.640e-01
<i>Prevotella copri</i>	74.2	Control	34	2.200e-02
<i>Bifidobacterium bifidum</i>	57.2	UC	70	1.130e-01
<i>Parabacteroides merdae</i>	63.2	Control	97	3.110e-01
<i>Lachnospiraceae bacterium 8 1 57FAA</i>	57.5	Control	185	8.790e-01
<i>Eubacterium eligens</i>	36.8	UC	43	4.000e-02
<i>Alistipes shahii</i>	42.5	Control	2	1.000e-03
<i>Bacteroides dorei</i>	65.2	UC	122	5.190e-01
<i>Collinsella aerofaciens</i>	58.5	UC	37	2.600e-02
<i>Dorea longicatena</i>	63.5	Control	27	9.000e-03
<i>Bacteroidales bacterium ph8</i>	41.2	Control	17	1.000e-03
<i>Eubacterium ramulus</i>	78.0	Control	5	1.000e-03
<i>Parabacteroides unclassified</i>	40.5	UC	173	7.500e-01
<i>Lachnospiraceae bacterium 2 1 58FAA</i>	61.3	Control	40	3.000e-02
<i>Clostridium boltea</i>	45.8	CD	29	1.000e-02
<i>Bacteroides vulgatus</i>	33.5	UC	177	8.050e-01
<i>Ruminococcus bromii</i>	43.2	Control	22	6.000e-03
<i>Roseburia inulinivorans</i>	47.2	Control	156	6.490e-01
<i>Oscillibacter unclassified</i>	52.5	Control	81	2.100e-01
<i>Coprococcus comes</i>	54.0	Control	38	2.700e-02
<i>Ruminococcaceae bacterium D16</i>	42.2	UC	116	5.050e-01

Figure 5. HTML output of aggregated ranked list for microbes predictive in PRISM dataset. CD, Crohn's disease; UC, ulcerative colitis.

Evaluation Over Training Data					
Model	AUC	MCC	Precision	Recall	F1
RF	1.00	1.00	1.00	1.00	1.00
SVM	0.92	0.72	0.83	0.82	0.82
Logistic Regression	0.95	0.77	0.85	0.85	0.85
MLPNN	1.00	0.95	0.97	0.97	0.97

Evaluation On External Data					
Model	AUC	MCC	Precision	Recall	F1
RF	0.73	0.19	0.36	0.45	0.37
SVM	0.75	0.36	0.60	0.57	0.57
Logistic Regression	0.73	0.26	0.52	0.51	0.50
MLPNN	0.71	0.19	0.47	0.46	0.46

Figure 6. HTML output of training and external evaluation for the PRISM dataset. RF, Random Forest; SVM, Support Vector Machine; MLPNN, multiple-layer perceptron neural network; AUC, area under the receiver operating curve; MCC, ; F1, .

Data availability

Source data

Original data and information for both the PRISM and external IBD datasets can be obtained from *Dataset 4* of the original

study <http://doi.org/10.1038/s41564-018-0306-4>³³. The data for the PRISM and external IBD datasets as formatted for use of Meta-Signer can be found at Zenodo: derekreiman/Meta-Signer: Original Release. <http://doi.org/10.5281/zenodo.4077403>³⁶.

This project contains microbial abundance and sample class data formatted for Meta-Signer as TSV, TXT and JSON files within the folder 'data'.

1. *abundance.tsv* - tab separated file for microbial abundance data
2. *labels.txt* - text file of class labels
3. *model_parameters.json* - JSON file containing neural network hyper-parameters

Extended data

Zenodo: derekreiman/Meta-Signer: Original Release. <http://doi.org/10.5281/zenodo.4077403>³⁶. This project contains the following extended data:

- ExtendedData.pdf (Data for the benchmarking of Meta-Signer against Biosigner and PERMANOVA ranked features)

Data are under the GPL-3 license.

Software availability

- Source code available from: <https://github.com/YDaiLab/Meta-Signer>
- Archived source code at time of publication: <http://doi.org/10.5281/zenodo.4077403>³⁶
- License: **GPL-3.0**

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

1. Marchesi JR, Adams DH, Fava F, *et al.*: **The gut microbiota and host health: a new clinical frontier.** *Gut*. 2016; **65**(2): 330–339.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
2. Wang J, Jia H: **Metagenome-wide association studies: fine-mining the microbiome.** *Nat Rev Microbiol*. 2016; **14**(8): 508–22.
[PubMed Abstract](#) | [Publisher Full Text](#)
3. NIH Human Microbiome Portfolio Analysis Team: **A review of 10 years of human microbiome research activities at the US National Institutes of Health, Fiscal Years 2007-2016.** *Microbiome*. 2019; **7**(1): 31.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
4. Wang J, Kurilshikov A, Radjabzadeh D, *et al.*: **Meta-analysis of human genome-microbiome association studies: the MiBioGen consortium initiative.** *Microbiome*. 2018; **6**(1): 101.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
5. Qin J, Li Y, Cai Z, *et al.*: **A metagenome-wide association study of gut microbiota in type 2 diabetes.** *Nature*. 2012; **490**(7418): 55–60.
[PubMed Abstract](#) | [Publisher Full Text](#)
6. Hale VL, Chen J, Johnson J, *et al.*: **Shifts in the Fecal Microbiota Associated with Adenomatous Polyps.** *Cancer Epidemiol Biomarkers Prev*. 2017; **26**(1): 85–94.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
7. Pascal V, Pozuelo M, Borruel N, *et al.*: **A microbial signature for Crohn's disease.** *Gut*. 2017; **66**(5): 813–822.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
8. Metwally AA, Yang J, Ascoli C, *et al.*: **MetaLONDA: a flexible R package for identifying time intervals of differentially abundant features in metagenomic longitudinal studies.** *Microbiome*. 2018; **6**(1): 32.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
9. Xia Y, Sun J: **Hypothesis Testing and Statistical Analysis of Microbiome.** *Genes Dis*. 2017; **4**(3): 138–148.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
10. Zhao N, Chen J, Carroll IM, *et al.*: **Testing in Microbiome-Profiling Studies with MiRKAT, the Microbiome Regression-Based Kernel Association Test.** *Am J Hum Genet*. 2015; **96**(5): 797–807.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
11. Wu C, Chen J, Kim J, *et al.*: **An adaptive association test for microbiome data.** *Genome Med*. 2016; **8**(1): 56.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
12. Wang T, Zhao H: **Constructing predictive microbial signatures at multiple taxonomic levels.** *J Am Stat Assoc*. 2017; **112**(519): 1022–1031.
[Publisher Full Text](#)
13. Koh H, Blaser MJ, Li H: **A powerful microbiome-based association test and a microbial taxa discovery framework for comprehensive association mapping.** *Microbiome*. 2017; **5**(1): 45.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
14. Hu J, Koh H, He L, *et al.*: **A two-stage microbial association mapping framework with advanced FDR control.** *Microbiome*. 2018; **6**(1): 131.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
15. Knights D, Parfrey LW, Zaneveld J, *et al.*: **Human-associated microbial signatures: examining their predictive value.** *Cell Host Microbe*. 2011; **10**(4): 292–296.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
16. Ditzler G, Polikar R, Rosen G, *et al.*: **Multi-Layer and Recursive Neural Networks for Metagenomic Classification.** *IEEE Trans Nanobioscience*. 2015; **14**(6): 608–616.
[PubMed Abstract](#) | [Publisher Full Text](#)
17. Pasolli E, Truong DT, Malik F, *et al.*: **Machine Learning Meta-analysis of Large Metagenomic Datasets: Tools and Biological Insights.** *PLoS Comput Biol*. 2016; **12**(7): e1004977.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
18. Fioravanti D, Giarratano Y, Maggio V, *et al.*: **Phylogenetic convolutional neural networks in metagenomics.** *BMC Bioinformatics*. 2018; **19**(Suppl 2): 49.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
19. Reiman D, Metwally A, Dai Y: **Using convolutional neural networks to explore the microbiome.** *Annu Int Conf IEEE Eng Med Biol Soc*. 2017; **2017**: 4269–4272.
[PubMed Abstract](#) | [Publisher Full Text](#)
20. Oudah M, Henschel A: **Taxonomy-aware feature engineering for microbiome classification.** *BMC Bioinformatics*. 2018; **19**(1): 227.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
21. Metwally AA, Yu PS, Reiman D, *et al.*: **Utilizing longitudinal microbiome taxonomic profiles to predict food allergy via Long Short-Term Memory networks.** *PLoS Comput Biol*. 2019; **15**(2): e1006693.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
22. Reiman D, Metwally AA, Sun J, *et al.*: **PopPhy-CNN: A Phylogenetic Tree Embedded Architecture for Convolutional Neural Networks to Predict Host Phenotype From Metagenomic Data.** *IEEE J Biomed Health Inform*. 2020; **24**(10): 2993–3001.
[PubMed Abstract](#) | [Publisher Full Text](#)
23. LaPierre N, Ju CJT, Zhou G, *et al.*: **MetaPheno: A critical evaluation of deep learning and machine learning in metagenome-based disease prediction.** *Methods*. 2019; **166**: 74–82.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
24. Zhou YH, Gallins P: **A Review and Tutorial of Machine Learning Methods for Microbiome Host Trait Prediction.** *Front Genet*. 2019; **10**: 579.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
25. Zhang Q, Abel H, Wells A, *et al.*: **Selection of models for the analysis of risk-factor trees: leveraging biological knowledge to mine large sets of risk factors with application to microbiome data.** *Bioinformatics*. 2015; **31**(10): 1607–1613.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
26. Pihur V, Datta S, Datta S: **RankAggreg, an R package for weighted rank aggregation.** *BMC Bioinformatics*. 2009; **10**: 62.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
27. Ho TK: **Random decision forests.** *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) -Volume 1*. 1995.
[Publisher Full Text](#)
28. Cortes C, Vapnik V: **Support-vector networks.** *Mach Learn*. 1995; **20**(3):

- 273–297.
[Publisher Full Text](#)
29. Hastie T, Tibshirani R, Friedman J, *et al.*: **The elements of statistical learning: data mining, inference, and prediction**. New York: Springer. 2009.
[Reference Source](#)
30. Aggarwal CC: **Neural networks in deep learning**. Springer. 2018.
[Reference Source](#)
31. Danaee P, Ghaeini R, Hendrix DA: **A deep learning approach for cancer detection and relevant gene identification**. *Pac Symp Biocomput*. 2017; **22**: 219–229.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
32. Anderson MJ: **Permutational multivariate analysis of variance (permanova)**. In: N. Balakrishnan *et al.*, editors, *In Wiley StatsRef: Statistics Reference Online*. Wiley, 2017; 1–15.
[Publisher Full Text](#)
33. Franzosa EA, Sirota-Madi A, Avila-Pacheco J, *et al.*: **Gut microbiome structure and metabolic activity in inflammatory bowel disease**. *Nat Microbiol*. 2019; **4**(2): 293–305.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
34. Tigchelaar EF, Zhernakova A, Dekens JAM, *et al.*: **Cohort profile: LifeLines DEEP, a prospective, general population cohort study in the northern Netherlands: study design and baseline characteristics**. *BMJ Open*. 2015; **5**(8): e006772.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
35. Rinaudo P, Boudah S, Junot C, *et al.*: **biosigner: A New Method for the Discovery of Significant Molecular Signatures from Omics Data**. *Front Mol Biosci*. 2016; **3**: 26.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
36. Reiman D, DaiLab Y: **derekreiman/meta-signer: Original release**. 2020.
<http://www.doi.org/10.5281/zenodo.4077403>

Open Peer Review

Current Peer Review Status:

Version 1

Reviewer Report 27 July 2021

<https://doi.org/10.5256/f1000research.30263.r87185>

© 2021 JI B. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Boyang JI

Department of Biology and Biological Engineering, Chalmers University of Technology, Gothenburg, Sweden

In the article “Meta-Signer: Metagenomic Signature Identifier based on rank aggregation of features”, Reiman et.al. developed a pipeline for identifying microbial features linked to human health/disease using multiple machine learning methods with rank aggregation of feature lists. Four different methods for classifications including random forest, support vector machine, logistic regression and multiple-layer perceptron neural network were applied. Also authors applied the pipeline with a dataset including healthy controls and IBD patients.

Overall, the pipeline is well-designed, and the paper is well written. Also this software represents an alternative way to identify microbial signatures linked to disease. However, I found that there are some points that could be analyzed and discussed more fully.

Major points:

1. In this workflow, the input abundance data were pre-processed, filtering and normalized by log transformation. More details about the pre-processing and filtering need to be included in the manuscript. Also lot of evidences (Weiss et al. Microbiome (2017) 5:27 DOI 10.1186/s40168-017-0237-y; Pereira et al. BMC Genomics (2018) 19:274 DOI 10.1186/s12864-018-4637-6; McKnight et al. Methods Ecol. Evol. (2018) 1–12 DOI 10.1111/2041-210X.13115) had shown that the methods of normalization had significant influences in the analysis of metagenome abundance data. Therefore, the use of log transformation of relative abundance may be not a good choice here. Authors can add more choices for normalization or just provide the direct functions for importing normalized abundance data.
2. Although authors provide the visualization of the relationship between number of features and the model performances (e.g. AUC) in the script of “generate_feature_ranking.py”, I still think authors can provide a completely automatic way for machine learning parameter optimization and the selection of best number of features.

Minor point:

1. Since the configuration file is a python file, I strongly suggest authors to change current

"config.py" to other readable names.

2. Some spelling errors in the title and main text

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: microbiome; omics integration; mathematical modeling; human disease; systems biology; bioinformatics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 29 June 2021

<https://doi.org/10.5256/f1000research.30263.r86264>

© 2021 Tierney B. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Braden T Tierney

Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA

The authors present a machine-learning software tool designed for use on compositional, specifically microbiome, data. This is a worthy endeavor -- the microbiome field is in need of well-designed statistical methods. I am specifically enthusiastic about using multiple machine learning approaches on the same dataset.

However, I do not believe this software is ready for wide use. The lofty goal of comparing multiple machine learning tools sets an extremely high bar for usability, especially to those in the microbiome field who are not machine learning experts. Its complexity, while impressive and indicative of substantial effort, requires that the authors spend a fair amount of time attempting to make its specific use-cases and pitfalls clear to users.

Additionally, I believe the manuscript itself needs substantially more detail in order to inform users as to what is going on under the hood. If I had to choose, I would like to see less time spent on the theory underlying each method and more on the justification and interpretation of the model input/output. As a result, I would recommend a major revisions to the tool and manuscript before its widespread adoptions.

Major Comments:

1. Software structure

1. I was unable to locate any evidence of a unit testing suite -- for a tool this complex, this is essential. If the authors do not have experience with professional-level Pythonic development, I highly recommend that they take some time and look into the fundamentals of unit-testing and writing object-oriented code.
2. Users should have a clear statement in the README of when the software is appropriate and when it is not
3. The example in the extended data of the manuscript should ideally be featured in the documentation (referenced in the README) so users have a clear use-case right on hand.
4. What about data on runtime or recommended sample sizes for different algorithms? What about memory requirements? A deep network has very different requirements than logistic regression -- this should be made clear in the documentation. You really do not want to accidentally encourage improper use of these algorithms -- eg using deep learning on a tiny dataset will yield overfitting.
5. On this note, I am not entirely sure deep learning should even be enabled by default. The sample sizes required for this to be meaningful are orders of magnitude larger than most microbiome studies currently available (certainly the ones used in the IBD example). Moreover, the power of deep learning when used appropriately comes from its customizability -- number of layers, window sizes, etc. I am not clear that it is an algorithm optimized for a "one click solution." I would rather see it replaced with, perhaps, a gradient boosted machine or something similar.
6. Parameters -- The authors are aggregating vastly dissimilar machine learning techniques. Again, this is a great sentiment, but each of them has so many different drawbacks and potential strategies that the authors need to clarify and justify their default choices in the GitHub repository. The config.py file has only a small subset of the options. Perhaps they could argue that Meta-Signer is a "first pass" tool, and that it could later be used to select an optimal method for refinement?

7. Please provide clear resources for the users to interpret your output (e.g. a model and an AUC is not enough). What do you recommend they do with the information they now have?

The manuscript leaves a series of very basic questions on the table.

1. Who is the target audience for this? How much of a machine learning background should they have? This perhaps sounds pedantic, but it is important. A machine learning expert would likely fit their own models to have better control over the data. A novice would likely not be able to use the software without more detailed documentation or -- worse -- they would use it and not fully understand what is going on technically.
 2. Why is this specifically targeted towards microbiome data? Why not other compositional data? Or can you use count data instead?
 3. Why were these particular modeling strategies included? Why not others?
 4. Can you discuss why you are distinct from other tools -- like BioSigner -- and why someone would use yours over that? Again, I would really like to see the results from this comparison actually discussed in depth. It is not sufficient to just have a pdf on the GitHub. Please discuss why you saw the differences you did.
 5. Can you include other potential demographic variables or confounders? We now know the microbiome is wildly confounded by environmental variables -- any modeling approach that does not explicitly discuss these should be treated with suspicion. See <https://pubmed.ncbi.nlm.nih.gov/33149306/>¹ or <https://www.nature.com/articles/nature15766>². Simply predicting a Y with a series of X's is not likely to be reproducible at scale.
 6. It is critical that the discussion has a clear mention of the benefits and drawbacks of this approach? Can you discuss overfitting at least? An AUC of 1 on your training dataset is likely too good to be true (and this was seen to be the case in the validation cohort, Figure 6).
- In summary, I do believe there is value to this tool -- but the authors need to exert more effort to clarify its utility and, most importantly, ensure it can be used to uncover meaningful biology.

References

1. Vujkovic-Cvijin I, Sklar J, Jiang L, Natarajan L, et al.: Host variables confound gut microbiota studies of human disease. *Nature*. **587** (7834): 448-454 [PubMed Abstract](#) | [Publisher Full Text](#)
2. Forslund K, Hildebrand F, Nielsen T, Falony G, et al.: Disentangling type 2 diabetes and metformin treatment signatures in the human gut microbiota. *Nature*. 2015; **528** (7581): 262-266 [PubMed Abstract](#) | [Publisher Full Text](#)

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

No

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

No

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

No

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: microbiology, metagenomics, bioinformatics, data science

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research