Gene Expression

ExpressionExpertIpynb: Promoter library analysis with a Jupyter Notebook machine learning workflow

Ulf W. Liebal[1,*], and Lars Blank[1]

[1] Institute of Applied Microbiology-iAMB, Aachen Biology and Biotechnology-ABBt, RWTH Aachen University, Worringerweg 1, 52074 Aachen, Germany

*To whom correspondence should be addressed.

Abstract

Motivation: In metabolic engineering enzyme expression is adjusted to rearrange metabolic pathways. Among the main determinants of gene expression are consensus sequences in and around the promoter region. However, the effects of consensus sequences depend on environmental context, may vary across organisms and are divergent for different sigma factors. Here, we present a workflow to support in the identification of sequence features with effects on gene expression.

Results: The workflow is based on a Python Jupyter Notebook and covers (i) the statistical sequence analysis, (ii) machine learning regression of sequence and GC-content to expression values, (iii) performance evaluation of the regressor, (iv) sampling of the theoretical exploration space, and (v) generation of novel sequences with defined activity. The notebook is flexible to also analyze multiple promoter libraries across multiple species to predict defined cross-host activity. It can be adapted to investigate any sequence library to identify sequence factors that determine targeted quantitative outcomes.

Availability: Freely available for git cloning at https://git.rwth-aachen.de/ulf.liebal/exp2.ipynb and licensed under the terms of GPLv3.

Contact: ulf.liebal@rwth-aachen.de

Supplementary information: Supplementary data available in the Git folder.

1 Introduction

Metabolic engineering requires the controlled adjustment of gene expression to optimize metabolite production. The regulation of gene expression can occur on several levels, and among the most well-known and commonly implemented control points is the promoter region. Sequences in the vicinity of the promoter region contain consensus sequences that determine expression activity. Despite the general characterization of the effects of consensus sequences, their effects may change in different environmental contexts, may vary across organisms and are divergent for different sigma factors. In this context promoter libraries are useful to get access to promoters over a range of activities and to identify sequence features responsible for activity[1–3].

Most promoter library studies have performed detailed machine learning analysis to identify sequence features that drive expression activity. However, the analyses are based on diverse programming tools implementing different algorithms and thus direct comparison of library analysis results is impeded and the reconfiguration of the analysis to new libraries is hampered. To unify the analysis and to provide a platform suitable for easy reconfiguration with present the ExpressionExpertIpynb tool. The ExpressionExpertIpynb is based on Python Jupyter Notebooks and consists of workflows to I) support in generating statistical overview to the promoter libraries, II) train machine learning algorithms to predict activities and identify sequence features affecting gene expression, III) predict novel sequences with defined activity. Jupyter Notebooks provide a versatile coding environment and allows users with basic programming experience to amend and expand the existing workflow to specific data needs. In the following, we show the application of the ExpressionExpertIpynb on a promoter library with lumped transcription factors in Bacillus subtilis[4] and an expression study of GFP and mOrange in Geobacillus[5].

2 Workflow and implementation

We aimed at realizing the guidelines for Jupyter Notebook program development[6] and all user input and configuration parameters are stored in the configuration file 'config.txt.'

2.1 Data preparation and statistical analysis

The first Notebook called '1-Statistical-Analysis' provides an overview to the promoter library and generates several statistical information regarding the sequence and the activity measurements. For some datasets only the average activity and standard deviation is available, however, machine learning algorithms cannot process statistical information and perform better if separate samples are available. Therefore, the workflow enables the generation of multiple samples based on mean and standard deviation.

This is a reasonable procedure if the original data was normal distributed without outliers. This procedure is beneficial if the sampling size is low (e.g. < 500 samples), increases the training robustness significantly while adding a reasonable prediction bias. The sequences are converted into a one-hot encoding.

The statistical analysis provides an overview to important properties of the promoter library, namely:

- sequence diversity
- position diversity
- nucleotide-position expression statistics
- expression strength distribution
- cross host expression strength

The promoter diversity is visualized by the sequence and position diversity. The sequence diversity is important to evaluate how well the sequence space was sampled. It is visualized by a histogram of the nucleotide exchange ratio relative to a reference or among all sequences. The position diversity informs about how many nucleotides have been sampled for each position. It is visualized as a stacked bar plot with the amount of each nucleotide on each position, and with the position specific entropy in a bar plot. The nucleotide-position expression statistics is calculated for all measurements on a specific nucleotide-position. For each sequence a matrix of one-hot encodings with the positions on the columns and the nucleotides on the rows was multiplied by the expression strength yielding a matrix with the expression strength at correct nucleotide-positions in the matrix. The matrices allowed calculation of average and the variance of expression on each nucleotide position and the corresponding heat-map shows which positions are associated with higher or lower expression. The expression strength is visualized with a histogram and a scatter plot for cross host expression.

2.2 Regressor training and performance

The implemented machine learning routines are random forest (RFR), gradient boosting (GBR), and support vector regression (SVR). The input features are nucleotides on each position and the overall sequence GC-content, and the target variable is the expression strength. The data is split into training and test set, with a default ratio of 9:1. The data is standardized with zero mean and unit variance for SVR, whereas tree-based methods perform well on original data. The library is scanned for uninformative positions with low position diversity and the user can adjust a parameter to exclude positions based on an entropy cut-off. A grid search identifies the optimal hyperparameters. The training is performed with cross-validation, by default with data separation in 9:1, and, a group shuffle split based on the sequences distributes replicates either to the test or the training set. The group shuffle split ensures that the test sequences are unknown to the regressor. The performance evaluation is based on the R^2-score and plot for the correlation of measured and predicted expression for training and test set is generated. The tree-based methods allow the extraction of feature importance and represent the contributions of each nucleotide-position to the prediction. They are extracted and visualized with a Logo-plot[7].

2.3 Synthetic sequence characterization

The workflow allows to identify the exploration space covered by the samples for which reliable prediction are possible and to test novel sequences within the exploration space. The exploration space spans the sequences for which reasonable predictions can be expected and is bounded by the sequence and position diversity. Only for sufficiently sampled nucleotides at any position can the regressor derive activity rules and predict reasonable activities. In addition, the sequence distance of an unknown sequence with respect to the reference sequence of all other sequences should be lower than the sequence distances in the training samples because unknown regulatory or physiological events can interfere with sequences deviating too much from the samples. The user defines the cut-off for the exploration space based on the visualizations of position and sequence diversities in the statistical analysis. The sequences of the exploration space can be generated exhaustively or a random subset of defined size and results in a synthetic promoter library with predicted expression activities. The workflow allows to select synthetic sequences close to a target expression and the synthetic library can be statistically analyzed similarly to the experimental library.

3 Performance

In the following, the results of two published examples are presented, available also in the ExpressionExpertIpynb package. The first example represents a single library measured for GFP in *Bacillus subtilis*[4]. The data contains 206 different sequences with 52 nucleotide positions recognized by various sigma factors analyzed collectively. The GFP mean and standard deviation is reported along with the information triplicate measurements, which was used to re-generate triplicates analytically. The main results from the ExpressionExpertIpynb workflow are shown in Figure 1 and Table 1. The Table 1 reports for the three regression strategies of random forest regression (RFR), gradient boost regression (GBR), and support vector regression (SVR) the regression run time in s (AMD Ryzen 5 2400G, 8-cores, 6MB RAM), the correlation coefficient over cross-validation sets, and the correlation coefficients of a randomly selected training-test-set separation. A feature importance for each nucleotide and the GC-content can be derived from the tree-based methods of RFR and GBR, and the corresponding importance for the GC-content is indicated. The corresponding data is visually presented in Figure 1 and organized into three rows: the top row representing sequence related statistical information, the middle row containing reporter protein associated information and the bottom row with the performance of RFR, GBR, and SVR. Two graphs on the very bottom indicate the feature importance results derived from the tree-based methods RFR and GBR. Whereas the training set is reasonably reproduced, and the feature importance of nucleotide positions reproduce known sequence determinants, *e.g.*, the G between –10 and –20, the test set is predicted not better than by chance.

The workflow in ExpressionExpertIpynb offers the feature to analyze parallel experiments involving two hosts or two reporter proteins for a reference promoter library. This feature is shown on a dataset on Geobacillus with a library containing 81 promoter sequences stretching 104 nucleotides where each promoter is measured with GFP and mOrange[5]. The difference to the single observation is represented in Figure 2, in the second row containing the statistical information of the two observations of reporter proteins. The scatter plot represents the conservation of expression strength among the reporter proteins and shows that no correlation of expression activity exists between the reporter proteins. The heat map indicates the difference between the average expression between GFP and mOrange for each nucleotide position. The strongest difference is detected for base 'C' on position –96, and a closer investigation of the effect of this position on the differential expression is warranted. The predictive potential of the machine learning regressors are mediocre, the correlation coefficients of the cross-validation are equal to random choices.

Table 1. ML Regression for the *Bacillus subtilis* library[4], see also Figure 1. The library is composed of 206 promoters in triplicates with changes in 52 nucleotides upstream of the transcription start site and measured for GFP activity. RFR: random forest regression, GBR: gradient boosting regression, SVR: support vector regression.

ML Type	RFR	GBR	SVR
Run Time (s)	315	825	183
R2 Cross-Val	0.3 (+/-0.4)	0.28 (+/-0.45)	0.3 (+/-0.51)
R2 Training	0.88	0.83	0.43
R2 Test	0.04	-0.36	0.03
GC-Content	11th	11th	N.A.

Table 2. ML Regression for the *Geobacillus* library[5], see also Figure 2. The library is composed of 81 promoters with a median number of 3 replicates with changes in 104 nucleotides upstream of the transcription start site and measured for GFP and mOrange activity. RFR: random forest regression, GBR: gradient boosting regression, SVR: support vector regression.

ML Type	RFR	GBR	SVR
Run Time (s)	275/264	712/719	94/97
R2 Cross-Val	-1.1 (+/-8.7) -0.57 (+/-1.9)	-0.97 (+/-5.4) -0.28 (+/-1.1)	-0.28 (+/-0.3) -0.13 (+/-0.4)
R2 Training	0.92/0.91	0.88/0.34	-0.18/0.47
R2 Test	-0.05/-0.2	-0.03/-0.07	-0.42/-0.18
GC-Content	1st, 9th	4th, 5th	N.A.

4 Discussion and conclusion

Synthetic promoter libraries are increasingly constructed to facilitate promoter selection with defined activities. The ExpressionExpertIpynb workflow supports the analysis of promoter libraries to identify the sequence information that determines expression strength. The identification is based on statistical analysis of the average nucleotide-position associated expression as well as the training of different machine learning algorithms. Moreover, the analysis of more complex libraries with multiple readouts, like two reporter proteins, transcript and protein level, or cross host expression can be analyzed. The tool facilitates data exploration by providing a workflow from data preparation, regressor training and performance evaluation and testing of novel sequences within the exploration space. The implementation in a Jupyter notebook facilitates rapid implementation and the low level scripting allows for direct adaptation of the workflow to specific needs.

So far, the analysis of promoter libraries was conducted with scripts tailored to the data, obfuscating reproducibility and interpretability. The ExpressionExpertIpynb is a contribution to harmonize analytical workflows and to enable an easy start for new groups with promoter libraries. Other toolboxes exist to support machine learning like tpot[8] and the JADBIO[9] tools. The advantage of the ExpressionExpertIpynb is to be general enough to be used across different data sets in expression studies, but remaining domain specific to facilitate simple data integration.

Acknowledgements

The authors thank Salome Nies, Dario Neves, and Sebastian Köbbing for promoter library data. The authors acknowledge discussions with Birgitta Ebert, Tobias Alter, Bastian Kister.

Funding

U.W.L. received funding by the Excellence Initiative of the German federal and state governments ((DE-82)EXS-PF-PFSDS015). The laboratory of L.M.B. is partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—Exzellenzcluster 2186, 'The Fuel Science Center ID: 390919832.'

Conflict of Interest: none declared.

References

1. Rhodius, V. A. & Mutalik, V. K. Predicting strength and function for promoters of the Escherichia coli alternative sigma factor, σE. *Proc. Natl. Acad. Sci.* **107**, 2854–2859 (2010).
2. Cuperus, J. T. *et al.* Deep learning of the regulatory grammar of yeast 5′ untranslated regions from 500,000 random sequences. *Genome Res* **27**, 2015–2024 (2017).
3. Cambray, G., Guimaraes, J. C. & Arkin, A. P. Evaluation of 244,000 synthetic sequences reveals design principles to optimize translation in Escherichia coli. *Nat. Biotechnol.* **36**, 1005 (2018).
4. Liu, D. *et al.* Construction, model-based analysis, and characterization of a promoter library for fine-tuned gene expression in Bacillus subtilis. *ACS Synth Biol* **7**, 1785–1797 (2018).
5. Gilman, J. *et al.* Rapid, Heuristic Discovery and Design of Promoter Collections in Non-Model Microbes for Industrial Applications. *ACS Synth. Biol.* **8**, 1175–1186 (2019).
6. Rule, A. *et al.* Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. *PLoS Comp Biol* (2019) doi:10.1371/journal.pcbi.1007007.
7. Tareen, A. & Kinney, J. B. Logomaker: beautiful sequence logos in Python. *Bioinformatics* **36**, 2272–2274 (2019).
8. Le, T. T., Fu, W. & Moore, J. H. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics* **36**, 250–256 (2020).
9. Tsamardinos, I. *et al.* Just Add Data: Automated Predictive Modeling and BioSignature Discovery. *bioRxiv* (2020) doi:10.1101/2020.05.04.075747.

Figures

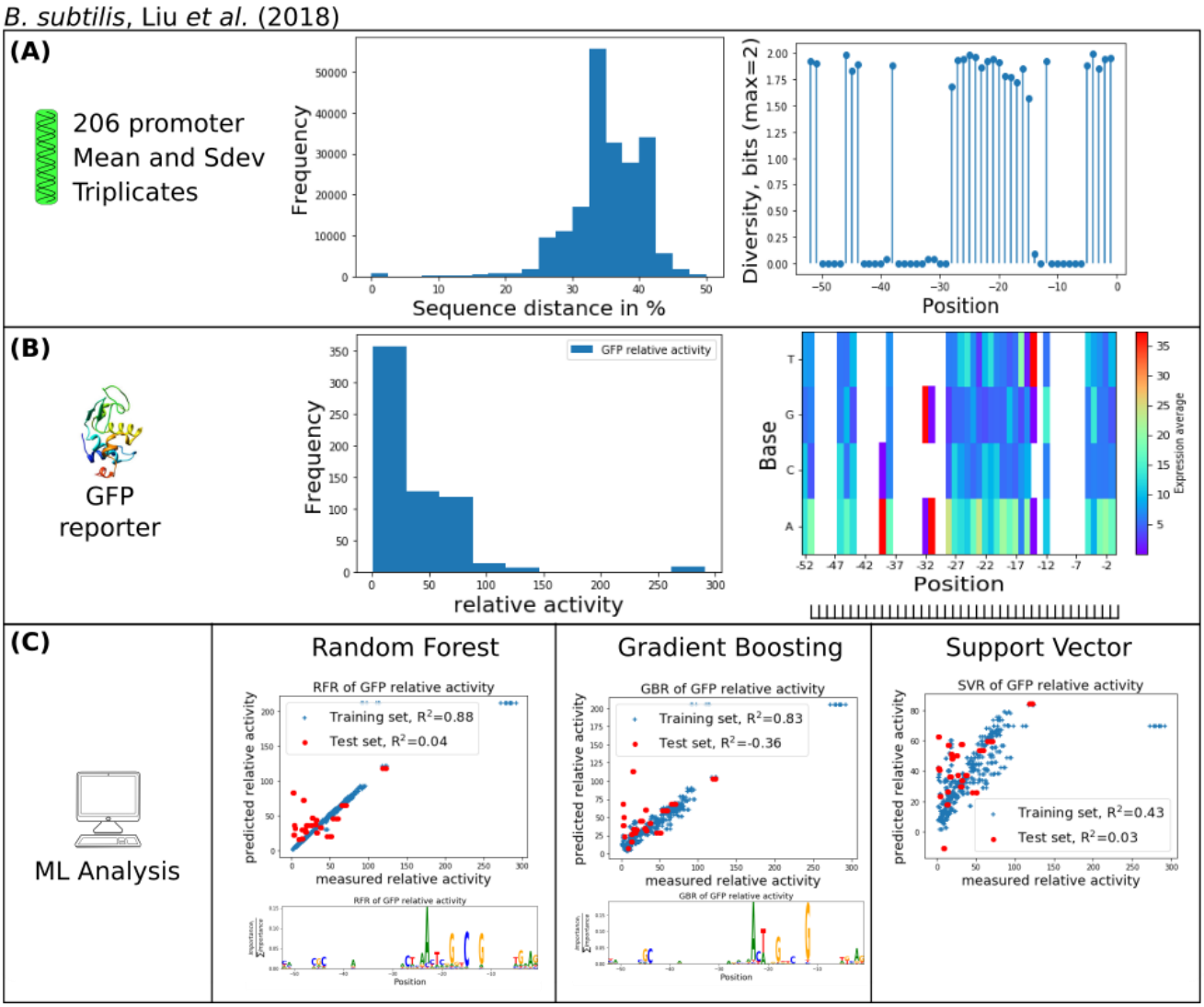

Figure 1: Figure 1: Analysis of B. subtilis library with 206 promoters over 52 nucleotides and activity measured by GFP in triplicates with reported mean and standard deviation (Liu et al., 2018). Panel (A) represents sample diversity indicators of sequence distance (pairwise nucleotide difference among all samples) and position diversity (entropy on each position with maximum at 2 bits). Panel (B) shows statistics related to the reporter protein GFP, namely the frequency count of relative activity and the mean expression associated with each nucleotide-position. Some positions, for example, -31[A, G] are associated with increased mean GFP activity. Panel (C) shows results of machine learning regression with random forest, gradient boosting and support vector machines. Whereas the training set is reasonably reproduced, the correlation coefficients indicate bad performance for the test set.

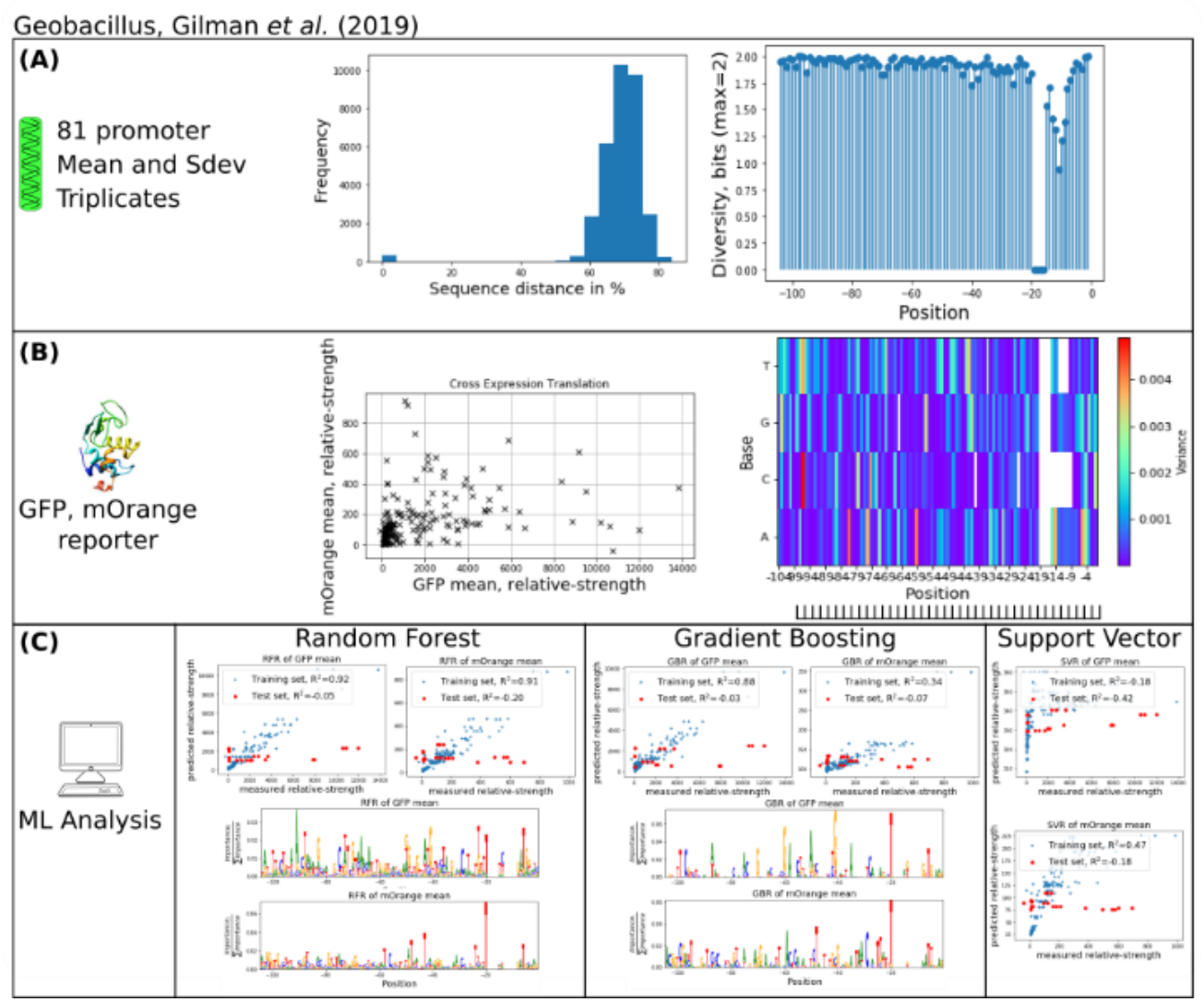

Figure 2: Analysis of Geobacillus library with 81 promoters on a sequence covering 104 nucleotides measured with the two reporter proteins GFP and mOrange (Gilman et al., 2019). Panel (A) shows statistical indicators of sample diversity similar to Figure 1. The statistics for the reporter proteins differs and the reporter activity histogram is replaced by a cross-output scatter plot, here for each promoter the relative strength of GFP and mOrange. The second plot is the variance of the mean of the nucleotide positions between GFP and mOrange corrected to show only significant differences of the mean as determined by Student's t-test. This metric indicates on which positions and nucleotides the expression is significantly different. Panel (B) shows results of the machine learning, and indicates that whereas the training set was reproduced reasonably (see also Table 2), the test set is badly reproduced.