Target-driven optimization of feature representation and model selection for microbiome sequencing data with *ritme*

Anja Adamov[1], Christian L. Müller[2,3,4], and Nicholas A. Bokulich[*1]

[1]Department of Health Sciences and Technology, ETH Zurich, 8092 Zürich, Switzerland
[2]Institute of Computational Biology, Helmholtz Zentrum München, 85764 Neuherberg, Germany
[3]Department of Statistics, Ludwig Maximilians University Munich, 80539 Munich, Germany
[4]Center for Computational Mathematics, Flatiron Institute, New York, NY 10010, USA

Abstract

Microbiome sequencing datasets are sparse, high-dimensional, compositional, and hierarchically structured. Predictive modelling from these data typically relies on *ad hoc* choices of feature representation, obscuring their impact on performance and biological interpretation. A standardized, compute-efficient framework is needed to jointly optimize microbial feature representation and model algorithms with transparent model evaluation. Here, we present *ritme*, an open-source software package implementing Combined Algorithm Selection and Hyperparameter Optimization tailored to microbial sequencing data. *ritme* systematically explores feature engineering methods — taxonomic aggregation, sparsity-aware selection, compositional transforms, and metadata enrichment — alongside diverse model classes using state-of-the-art optimizers and model trackers. Applied to three real-world use cases, *ritme* outperforms original study pipelines and generic AutoML baselines. It further provides users with insights

[*]Corresponding author: nicholas.bokulich@hest.ethz.ch

into how feature and model choices drive predictive performance. Together, these results establish *ritme* as a standardized framework for identifying optimal feature-model combinations from high-throughput sequencing data. *ritme* is an open-source Python package available at `https://github.com/adamovanja/ritme`.

Keywords: Microbiome, Automated machine learning, Compositional feature engineering, Multi-omics, Next-generation sequencing data

1 Introduction

Microbiomes are complex ecosystems of microorganisms and their functional components, which play key roles in human health, agriculture, and the environment. In human health, microbial dysbiosis has been implicated in inflammatory bowel disease [1] and in numerous immunological and metabolic conditions [2, 3]. In agriculture, microbiome composition governs soil health [4], wine quality [5], and crop performance [6]. In environmental systems, microbiomes play important roles in food chains as well as biogeochemical cycling [7], and shifts in microbial compositions are associated with temperature [8] and salinity gradients [9]. Discovering new functional roles and interactions within complex microbiomes can be challenging due to the high dimensionality as well as the large proportion of uncharacterized species and functions present in many ecoystems. Hence, identifying robust microbial signatures for these covariates typically involves training machine learning (ML) models on microbial features to predict selected targets such as disease states, environmental parameters or agronomic traits [10, 11]. In these models, effective feature representation and model choice are crucial for optimizing predictive performance and obtaining robust biological insights [12, 13, 14, 15], particularly given the limited sample sizes relative to the large number of features typical of microbiome datasets [16].

Standardized bioinformatics pipelines exist for processing raw microbial nucleotide sequence reads into feature count tables of similar sequence clusters [17, 18, 19, 20] or error-corrected reads [21]. However, there is a lack of user-friendly and optimized workflows for their downstream processing that address all statistical properties inherent to microbial count data - including sparsity, high dimensionality, compositionality [22], and phylogenetic and functional hierarchies - while optimizing predictive performance. Researchers often make subjective, non-reproducible decisions about microbial feature aggregation, selection, and transformation prior to model optimization, obscuring how these influence predictive performance and the biological

insights obtained [23, 16]. Alternatively, they address only selected statistical properties with microbiome-specific feature representation methods restricted to fixed, manually defined cutoff values evaluated via grid-search [15] or evaluated on a predefined predictive model algorithm and associated hyperparameters [24].

Many automated ML (AutoML) pipelines perform Combined Algorithm Selection and Hyper-Parameter Optimization (CASH) to optimally find the best model algorithm and feature preprocessing steps (e.g.,[25, 26, 27, 28]). Their feature processing repertoire is largely confined to generic statistical operations (e.g., constant variance feature removal and standardization [27]) and dimensionality reduction techniques (e.g., feature embeddings and matrix decompositions [28]), rather than domain-specific microbiome feature engineering. To date, no modelling framework exists that systematically explores and optimizes microbial feature representation in concert with model class and hyperparameters, addressing all statistical properties of microbial nucleotide sequencing data [23, 29].

Here, we introduce *ritme* (= Robust Inference of Feature Transformations and Model Ensembles for Next-Generation Sequencing Data), a user-friendly and open-source software package that systematically optimizes microbial feature representations and ML pipelines tailored to the statistical properties of microbial sequencing data. *ritme* includes extensive and customizable feature-model-space configurations and state-of-the-art optimization algorithms to efficiently search the vast hyperparameter space on distributed compute environments. Its modular architecture makes it scalable to future additions of other multi-omics modalities. Accessibility across diverse programmatic skill sets is ensured with *ritme*'s multiple user interfaces and automated experiment tracking and evaluations.

We demonstrate the utility and versatility of *ritme* on three real-world case studies: (i) predicting an infant's age from its gut microbiota, (ii) predicting ocean temperature from its microbiota, and (iii) predicting absolute microbial abundances from relative abundances. In each use case, *ritme* improves predictive performance compared to original study methodologies while providing insights into the contribution and relevance of individual modelling hyperparameters with regard to predictive performance. Together, these results underscore *ritme*'s potential to standardize the identification of best-performing feature-model combinations from high-throughput sequencing data, enabling more reproducible and robust biological discoveries.

2 Materials and Methods

2.1 Overview of main functions in *ritme*

The main goal of *ritme* is to optimally infer the best feature representation and model selection for a specific predictive task from microbial nucleotide sequencing data. To do this it offers three main functionalities (Figure 1) that fully adhere to best practices in performing ML analyses of microbiome data [29]:

- `split_train_test` merges the provided feature and metadata tables and randomly splits the merged data with the option to perform the split in a grouped manner - containing related sampled (e.g., from repeated measurements of same host or field site) within a single partition. This enables controlled training designs, e.g., leave-one-subject out or leave-one-study out training routines.

- `find_best_model_config` performs a systematic, optimized exploration of the feature-model parameter space on the training set (Section 2.2), implementing CASH across both the microbiome-specific feature engineering (Section 2.3) and predictive model options (Section 2.4). It further provides insights into contributions of each feature representation and model component to predictive performance.

- `evaluate_tuned_models` assesses the generalization capacity of optimal feature-model configurations per model type through performance evaluation (root mean squared error (RMSE), R^2, Pearson correlation, true-vs-predicted scatter plots) on both training and independent held-out test sets, thereby mitigating selection bias and preventing inflated performance estimates characteristic of the "winner's curse" phenomenon [23, 30].

2.2 Search for best feature-model configuration

ritme's main function `find_best_model_config` optimally searches for the best feature-model combination on the provided dataset. Each combination is derived from 80% of the provided training data and evaluated based on the root mean squared error (RMSE) of the remaining validation set, with the data being partitioned according to the defined grouping criteria (see `split_train_test` in Section 2.1). The combinations of feature engineering and model hyperparameters are iteratively evaluated per model type for a maximal run duration specified by the user (Figure 2). After completion,

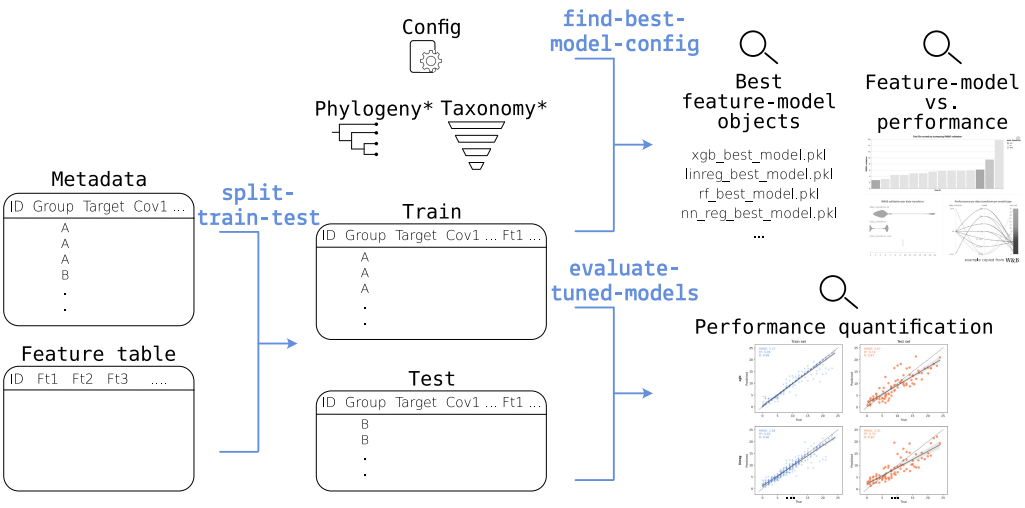

Figure 1: Overview of *ritme*'s three core functions: `split_train_test` merges the feature and metadata tables and partitions the data; `find_best_model_config` infers the optimal feature-model combination given an experiment configuration and provides insights into all explored combinations (inputs marked with an asterisk are optional); and `evaluate_tuned_models` quantifies predictive performance of the best feature-model combinations on the train and test sets.

`find_best_model_config` returns custom-made model objects `TunedModel` — each encapsulating the selected feature aggregation, selection, transformation, enrichment schema, and fitted estimator — thereby enabling reproducible, directly applicable prediction on any dataset with the same feature structure (Figure S1).

To perform the search for the best combined feature representation and modelling algorithm in a resource-aware manner, *ritme* leverages the Ray Tune framework [31] enabling optimal computational resource management with parallelisation across all available computational resources - from single workstations to high-performance computing environments. The Asynchronous Successive Halving Algorithm (ASHA) [32] and HyperBand early stopping algorithm [33] allow optimal scheduling of different hyperparameter configurations by terminating unpromising configurations early and adaptively allocating more resources to promising candidates, with the HyperBand algorithm providing deterministic scheduling when experiment reproducibility is needed (scheduling step in Figure 2).

To enable efficient hyperparameter search, hyperparameter suggestions are generated dynamically (define-by-run style) with different sampler algo-

rithms using Optuna [34] (sampling step in Figure 2), including the Tree-structured Parzen Estimator (TPE) [35, 36], the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [37, 38], a Gaussian process-based sampler (GPSampler) [34], a Quasi Monte Carlo Sampler (QMCSampler)[35, 39] and a random independent sampler [34]. For TPE, CMA-ES, and the GP sampler, the initial 1'000 trials are initialized with random configurations created with the random independent sampler.

The predictive performance of different launched hyperparameter combinations is trackable with two popular ML experiment tracking tools (tracking step in Figure 2): MLflow [40] and Weights and Biases [41], for each of which a reproducible example template is provided for optimally evaluating *ritme* experiments (Section 4).

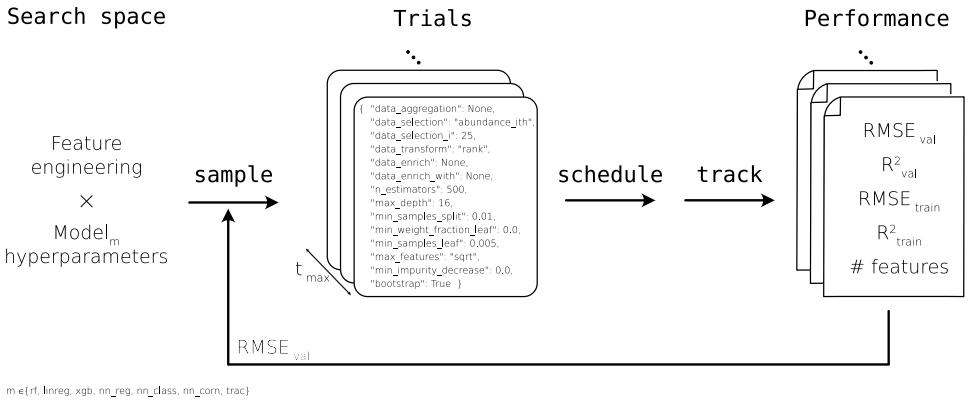

Figure 2: Schematic for how the optimal feature-model combination is inferred with *ritme*'s `find_best_model_config` (built on Ray Tune). For each model class m, trials are launched for a specified maximal run duration $t_{\max}$. In each trial, Optuna samplers propose a joint configuration over the microbiome-specific feature-engineering and model-hyperparameter spaces. Scheduling and early stopping are handled by ASHA or HyperBand. Trial metrics are tracked with MLflow or Weights & Biases and computed on an 80/20 split of the training data (trained on 80%, validated on 20%).

2.3 Microbiome-specific feature engineering

ritme includes feature engineering methods that account for diverse statistical properties of microbial sequencing data (Figure 3). Feature representation is comprised of four sequential steps: taxonomic aggregation, feature selection, compositional data transformations, and feature enrichment. The possible combinations of these steps, along with their associated parameters (i, q &

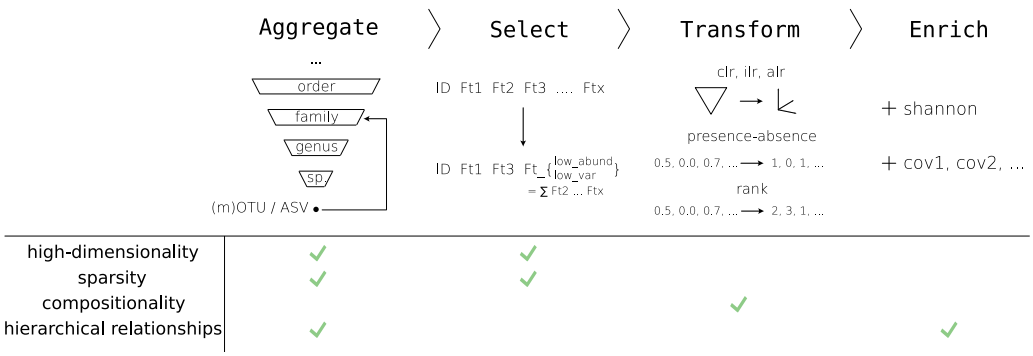

Figure 3: Four sequential microbiome-specific feature engineering steps implemented in *ritme* addressing the statistical properties of microbial sequencing data.

t, Table 1) and model-specific parameters (Section 2.4) are systematically evaluated as hyperparameters during optimization.

The taxonomic aggregation enables hierarchical aggregation of sequencing counts based on their taxonomic ranks where features are either left as-is (no aggregation) or summed according to rank. If the selected taxonomic entity is unknown for a specific sequence, the higher order rank is considered for aggregation.

To address the inherent sparsity and high-dimensionality of microbial sequencing data, *ritme* implements multiple feature selection strategies based on either abundance or variance criteria. These approaches systematically select features while summing the remaining ones into a single aggregate feature, preserving the compositional nature of the data. Four parametric selection methods are available:

- **Top-rank selection**: Selects the i most abundant/variant features while summing the remaining features (abundance_topi and variance_topi in Table 1).
- **Rank-referenced selection**: Selects features for inclusion based on comparison to the i-th ranked feature, summing those whose cumulative abundance/variance falls below this reference (abundance_ith and variance_ith in Table 1).
- **Quantile-based selection**: Selects features exceeding a specified quantile threshold in abundance/variance while summing features below this threshold (abundance_q and variance_q in Table 1).
- **Absolute threshold selection**: Selects features surpassing a defined

absolute threshold value in abundance/variance while summing the remainder (abundance_threshold and variance_threshold in Table 1).

Across all feature selection approaches, default hyperparameter ranges are inferred from the training data unless explicitly set by the user: i spans the integers $1, \ldots, p$, with p denoting the number of features in the training set, q is explored over the interval $[0.5, 0.9]$ and t ranges from the minimum to the maximum observed feature-wise summed abundance or variance.

ritme addresses the compositionality constraint through several log-ratio transformations, including centered (clr), isometric (ilr) and additive (alr) log ratio [42, 43, 44], a binary presence-absence transformation (pa) [45] and rank order normalization (rank) [46, 47]. For the additive log ratio transform, the feature with the highest relative abundance is selected as a reference.

Additionally, to allow for inclusion of contextual and microbiome-derived diversity information, *ritme* has the option to supplement the microbial features with Shannon diversity [48] and/or metadata fields.

The resulting search space of feature engineering options and associated hyperparameters can be restricted by the user in case the biological interest lies in a subset of the available representations. Additionally, the user can define the starting point(s) of the hyperparameter search.

Table 1: Number and values of categorical (cat) and conditional continuous (c-cont.) hyperparameters for each feature engineering method in *ritme*. The conditional hyperparameter values, depicted in brackets, are considered only when their corresponding categorical parent parameter is selected.

Name	**# cat**	**# c-cont.**	**Values**
Aggregation	1	0	None, tax_class, tax_order, tax_family, tax_genus
Selection	1	1	None
			abundance_topi (i), variance_topi (i)
			abundance_ith (i), variance_ith (i)
			abundance_quantile (q), variance_quantile (q)
			abundance_threshold (t), variance_threshold (t)
Transform	1	0	None, clr, ilr, alr, pa, rank
Enrichment	1	0	None, shannon, shannon_and_metadata, metadata_only

2.4 Predictive modelling frameworks

ritme implements a diverse set of ML models for regression tasks, spanning linear, ensemble, and gradient-based methods, and encompasses both parametric and non-parametric approaches (Tables 2 and 3). Each model is

evaluated based on its predictive performance on a validation set, as quantified by the root mean squared error (RMSE). For iterative gradient-based models, such as XGBoost and neural networks, training progress is monitored using MLflow [40] or Weights and Biases [41], with scheduler's early stopping mechanisms in place to optimize search efficiency (Section 2.2).

The neural network implementations in *ritme* are tailored to three learning tasks: regression `nn_reg`, classification `nn_class`, and ordinal regression `nn_corn` (Table 2). The ordinal regression neural network, CORN [49], is particularly suited for targets with a natural class order with the continuous target values being rounded to integers.

Each model is integrated with the feature engineering pipeline described in Section 2.3, facilitating systematic evaluation across diverse feature representation for each model. Only the trac model [50] contains a complementary approach by directly incorporating phylogenetic hierarchies and dynamically learning optimal taxonomic aggregation levels during training. Prior to modelling, features for linear regression are standardized using standard scaling, while neural networks utilize batch normalization [51] to normalize features before they are passed through the network. Similar to feature engineering hyperparameters, users can restrict the model hyperparameters tested and define the starting point(s) of the search.

Table 2: Predictive models implemented in *ritme*, with MSE = mean squared error, trac = tree-aggregation of compositional data, CORN = Conditional Ordinal Regression for Neural networks.

Name	**Algorithm**	**Implementation**	**Training Loss**
`linreg`	Elastic net linear regression	scikit-learn [52]	Elastic Net
`rf`	Random forest	scikit-learn	MSE
`trac`	trac [50]	adapted from c-lasso [53]	Squared error
`xgb`	XGBoost [54]	XGBoost [54]	Squared error
`nn_reg`	Neural network	PyTorch Lightning [55]	MSE
`nn_class`	Neural network	PyTorch Lightning	Cross-entropy
`nn_corn`	CORN [49]	coral-pytorch [49]	CORN loss [49]

2.5 Implementation and availability

ritme is an open-source Python 3 package distributed under the BSD 3-clause license, available via Anaconda at `https://anaconda.org/adamova/ritme`. It offers a user-friendly interface through both a Python API and a command-line interface, facilitated by the *Typer* library [56]. The *ritme*

Table 3: Number (#) and names of categorical (cat.) and continuous (cont.) hyperparameters for each predictive model in *ritme*. The neural network parameter n_units_hl$x$ represents the number of neurons in the x-th hidden layer, where n_hidden_layers specifies the total number of hidden layers. All remaining parameter names match those of the original package implementations. Asterisk-marked parameters (*) were categorized to allow None as a hyperparameter option. Dagger-marked parameters (†) were discretized to reduce the hyperparameter search space. Evaluated default ranges for each hyperparameter are listed in Table S1.

	#		**Hyperparameter names**	
Model	**cat**	**cont.**	**cat.**	**cont.**
`linreg`	0	2	-	alpha, l1_ratio
`rf`	3	5	max_depth*, max_features* bootstrap	n_estimators, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, min_impurity_decrease
`trac`	0	1	-	lambda
`xgb`	0	10	-	n_estimators, max_depth, min_child_weight, subsample, eta, num_parallel_tree, gamma, reg_alpha, reg_lambda, colsample_bytree
`nn_reg`, `nn_class`, `nn_corn`	n_hidden_layers + 3	5	n_units_hl$x$†, learning_rate†, batch_size†, epochs†	n_hidden_layers, dropout_rate, weight_decay, early_stopping_patience, early_stopping_min_delta

repository, accessible at `https://github.com/adamovanja/ritme`, includes comprehensive and reproducible tutorials and usage examples to assist users in running and evaluating *ritme* experiments.

ritme is implemented in a modular fashion facilitating future extensions with additional feature representation or modelling algorithms, enabling the open-source community to incorporate feature representations of other multi-omics modalities and domain-specific models. To ensure code reliability and robustness, *ritme* was developed in accordance with best practices in software development [57, 58], incorporating unit testing and continuous integration on Ubuntu and macOS.

2.6 Comparative modelling

2.6.1 *ritme* vs. original studies

To demonstrate the capabilities of *ritme*, we selected three real-world datasets which were all previously used to predict a continuous target from its microbial features (Table 4). For each use case, we initially split the data using the *ritme* `split_train_test` method in a train set and a test set, 80% and 20% of all samples. For use case 1, which includes repeated measurements per host, we applied a grouped split by host identifier to prevent information leakage between train and test sets. On the train set, we launched *ritme*'s `find_best_model_config` and additionally reproduced the original publication's feature engineering and modelling set-ups for comparison. The left-out test set was used to evaluate the predictive performance of the original approaches and the best feature-model configurations selected by *ritme* with the lowest RMSE on the validation set.

For all use cases, *ritme*'s `find_best_model_config` was launched for a duration of 119 hours per model type (Table 2). For the use cases 1 and 2 all model types were included. For the use case 3, `nn_class` and `nn_corn` were not considered due to logarithmic targets rounding to the same class and `trac` was excluded since the phylogeny could not be reproduced with missing raw sequencing data. The hyperparameter search spaces for all use cases were searched using the TPESampler.

Given the breadth of the feature-model search space and the moderate sample sizes of the selected use cases, the insights presented in Section 3 should be considered illustrative. Additional analyses are required to derive reliable biological conclusions.

Table 4: Datasets used for use cases comparing *ritme* to original models. Number of features (# features) include features that are non-zero in at least one sample. HTS = high-throughput sequencing, Ref. = reference, OTUs = operational taxonomic units, miTAGs = metagenomic 16S ribosomal RNA gene tags, abs. = absolute, transf. = transformed, mOTUs = marker gene based operational taxonomic units.

Use case	**Target**	**# Samples**	**# Features**	**Feature type**	**HTS method**	**Ref.**
U1	Infant age (months)	448	18'150	16S OTUs [17, 18]	Amplicon	[59]
U2	Ocean temperature (celsius)	139	35'651	16S OTUs derived from miTAGs[19, 60]	Shotgun	[8]
U3	Abs. microbial load (log10 transf.)	1'894	2'123	mOTUs [20]	Shotgun	[61]

2.6.2 *ritme* vs. generic autoML

To test whether microbiome-specific data engineering actually improves predictive performance, we selected the best performing model type identified by *ritme* and launched *auto-sklearn* [28] for the equal amount of time with the same amount of computational resources with its most comparative regressor (e.g., for `xgb` - *auto-sklearn*'s `gradient_boosting`, for `rf` - *auto-sklearn*'s `random_forest`). *auto-sklearn* was launched to also optimize the RMSE with a single best model (no ensembles were allowed) on the same training data and evaluated on the same held-out test set. *auto-sklearn* was selected since it is a state-of-the-art library for AutoML and was the top performer in previous AutoML challenges [62].

2.7 Data fetching and processing

2.7.1 Use case 1 - infant age

Sequencing data and associated metadata were fetched with *q2-fondue* [63] from the NCBI Sequencing Archive [64] using the reported BioProject ID PRJEB5482. Additional metadata were retrieved from the publication's [59] supplementary materials. According to the original publication, forward and reverse reads were trimmed to at most 162 nucleotide length using *q2-cutadapt* [65]. The overlapped paired reads were clustered sharing $\geq$ 97% similarity matched to the 13_8 99% Greengenes reference [66] using *q2-vsearch*

[67], with remaining reads clustered de novo. The resulting feature count table consists of 448 samples from 25 unique infants with 18'150 16S OTUs (operational taxonomic units) (Table 4).

Taxonomy was assigned by a trained naive Bayes classifier on the Greengenes 13_8 99% reference database using *q2-feature-classifier* [68], after extracting the V4 region. The phylogenetic tree was inferred by aligning sequences with MAFFT and masking the alignment using *q2-alignment* [69], constructing and midpoint-rooting the tree using FastTree in *q2-phylogeny* [70].

2.7.2 Use case 2 - ocean temperature

Sequencing data, associated metadata and sequence taxonomy was fetched from the original publication's [8] companion website `https://ocean-microbiome.embl.de/companion.html`. The resulting sequence feature count table consists of 139 samples with 35'651 16S OTUs, derived from metagenomic 16S ribosomal RNA gene tags (miTAGs) [19, 60] and clustered at 97% similarity to the release 115 SILVA 16S reference database [71] (Table 4).

The phylogenetic tree was inferred by aligning sequences with MAFFT and masking the alignment using *q2-alignment* [69], constructing and midpoint-rooting the tree using FastTree in *q2-phylogeny* [70].

2.7.3 Use case 3 - microbial load

Sequencing data, associated metadata and sequence taxonomy was fetched from the original publication's [61] GALAXY/MicrobLiver study population upload at `https://zenodo.org/records/14243685`. The resulting sequence feature count table consists of 1'894 samples with 2'123 marker gene based OTUs (mOTUs) [20] (Table 4).

3 Results

3.1 *ritme* outperforms original feature-model set-ups

For all three use cases (Table 4), the optimal feature-model configuration identified by *ritme* outperforms the original study configurations when evaluated on the same held-out test set (Tables 5 to 7). Moreover, *ritme*'s best models overfit less on the training data than the original study models (with training RMSE original vs. *ritme*: 1.01 vs. 1.08 in use case 1; 0.11 vs. 0.58 in use case 2; 0.05 vs. 0.07 in use case 3, Figures S2 to S4). For use case 3,

ritme's marginal test prediction improvement aligns with known constraints in ML-based prediction of absolute from relative abundances [72].

Consistently, *ritme*'s best feature-model configurations were different than the original approaches. As for the model algorithm, *ritme*'s selected model type overlaps with the original set-up only in use case 3. In terms of feature engineering, in two out of three use cases (use cases 2 and 3), *ritme* identified more parsimonious configurations with fewer features than the original setups as best performing. All original approaches used the most granular microbial feature count representation with no feature aggregation, whereas *ritme* selected taxonomic aggregations in two out of three use cases (use cases 2 and 3). Whereas the original studies employed fixed threshold-based feature selection criteria, *ritme* selected the microbial features based on their variance. Both modelling setups selected different strategies to address the compositionality of the data. Across all three use cases, *ritme* identified that enriching microbial features with metadata and diversity metrics — beyond what was employed in the original studies — improved predictive performance.

Table 5: Use case 1: Comparison of performance metrics on a held-out test set and configuration components between the original published approach [59] and *ritme*. For the original feature selection, the threshold is based on relative abundance values. Metadata used for feature enrichment with *ritme* includes the following variables at sampling: health status, milk diet, weaning status and antibiotics exposure 7 days prior. seq = sequences, R^2 = coefficient of determination, RMSE = root mean squared error.

		Original	***ritme***
Metrics	R^2	0.669	0.758
	RMSE	3.498	2.991
Config	Model algorithm	Random Forest [52]	`xgb`
	Feature aggregation	-	-
	Feature selection	$\geq 10^{-3}$ in ≥ 2 samples	variance_topi, $i = 6'873$
	Feature transform	Rarefaction to 2'000 seq/sample	ilr
	Feature enrichment	-	metadata_only
	# Features	850	6'879

Table 6: Use case 2: Comparison of performance metrics on a held-out test set and configuration components between the original published approach [8] and *ritme*. For the original feature selection, the threshold is based on relative abundance values. $x0$ represents a pseudocount = 1e-6.. CV = cross-validation, R^2 = coefficient of determination, RMSE = root mean squared error, pa = presence-absence transformation.

		Original	***ritme***
Metrics	R^2	0.887	0.900
	RMSE	2.321	2.189
Config	Model algorithm	Elastic Net [52] with 5-fold CV	`rf`
	Feature aggregation	-	tax_class
	Feature selection	$\geq 10^{-4}$ in any sample	variance_quantile, $q = 0.849$
	Feature transform	$log10(x + x0)$ & rank	pa
	Feature enrichment	-	metadata_only
	# Features	$7'250$	36

Table 7: Use case 3: Comparison of performance metrics on a held-out test set and configuration components between the original published approach [61] and *ritme* on GALAXY/MicrobLiver study population. For the original feature selection, the threshold is based on relative abundance values. $x0$ represents a pseudocount = 1e-4. The metadata enrichment in *ritme* was performed with the cohort feature. CV = cross-validation, prev. = prevalence, std. = standard, R^2 = coefficient of determination, RMSE = root mean squared error.

		Original	***ritme***
Metrics	R^2	0.434	0.444
	RMSE	0.148	0.147
Config	Model algorithm	XGBoost [54] with 10-fold CV	`xgb`
	Feature aggregation	-	tax_genus
	Feature selection	mean $> 10^{-4}$ & prev. $> 10^{-1}$	variance_ith, $i = 227$
	Feature transform	$log10(x + x0)$ & std. scale	-
	Feature enrichment	shannon	shannon_and_metadata
	# Features	446	331

3.2 *ritme* provides insights into optimal feature representation and model selection

During inference, each feature-model configuration selected by *ritme* is evaluated on the validation set (Section 2.2 and Figure 2). With *ritme*, users can inspect these selected configurations and their contributions to the predictive performance (Figure 1). Examining configurations across the three use cases (Figure 4) shows that validation performance distributions vary significantly by feature engineering approach and model type. Notably, the top-performing configuration identified for each use case (Tables 5 to 7) does not consistently represent the best median performance across trials. It is important to note that the displayed configurations also vary in model-specific hyperparameters within each category, which can confound the observed between-category differences.

By evaluating the number of features of the tested configurations compared to their predictive performance, *ritme* users additionally have the option to identify the model type (Figure 5a) or model trial (Figure 5b) with the minimal number of features with acceptable predictive performance.

Analysis of validation performance trajectories across trials reveals model-specific search effectiveness (Figure 6). All models except `linreg` exhibited substantial performance gains with TPE-guided sampling after the initial 1'000 random trials. Models with longer inference times (neural networks and XGBoost) completed fewer total trials. For such models, constraining hyperparameter ranges to smaller architectures may improve search efficiency. Users can experiment with alternative search algorithms through *ritme* to optimize exploration-exploitation balance for their specific datasets.

3.3 Microbiome-specific engineering enables *ritme* to outperform generic AutoML

Under matched compute budgets and identical train/test sets, we compared the best feature-model configurations identified by *ritme* to the generic baseline *auto-sklearn* (Section 2.6.2). In use cases 1 and 2, *ritme* achieved better predictive performance on the held-out test sets (Table 8 and Figures S2 and S3). These results indicate that microbiome-specific feature engineering is critical for optimally inferring predictive models from microbial feature count tables.

For use case 3, *ritme* performed comparably to *auto-sklearn* (Figure S4) and achieved marginal improvements over the original study (Table 7). This limited performance gain aligns with recent findings demonstrating constraints in predicting absolute from relative microbial abundances using ML-

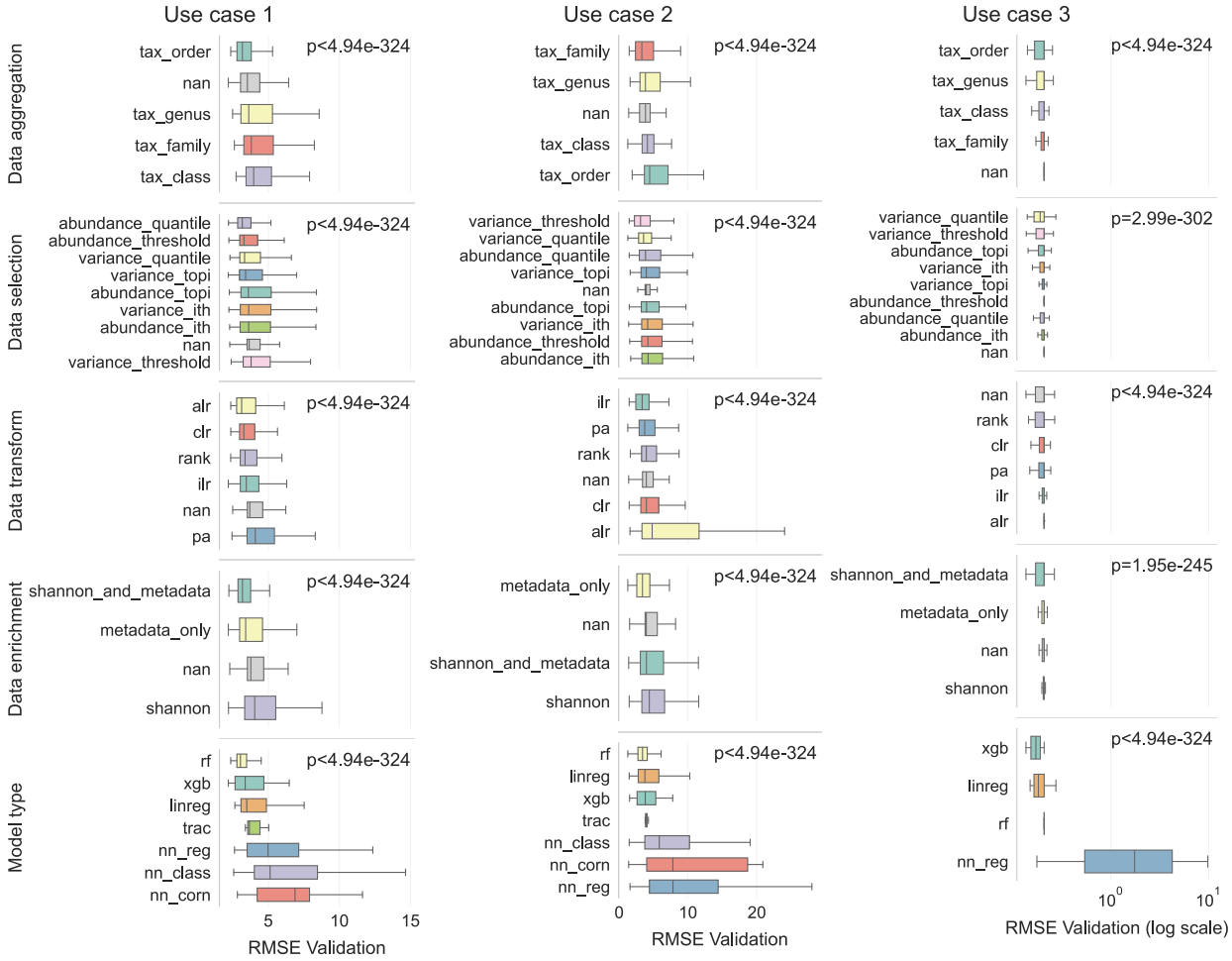

Figure 4: Distributions of validation performance across *ritme*'s tested feature engineering and model configurations quantified by validation root mean squared error (RMSE) for all three use cases sorted by increasing median RMSE; p-values report Kruskal–Wallis tests across groups.

based prediction approaches [72].

Table 8: Comparison of predictive performance of *ritme* and *auto-sklearn* on held-out test sets across the three use cases. Top performance indicated by boldface text. R^2 = coefficient of determination, RMSE = root mean squared error.

	$\mathbf{R^2}$		**RMSE**	
Use case	*ritme*	*auto-sklearn*	*ritme*	*auto-sklearn*
1	**0.758**	0.677	**2.991**	3.454
2	**0.900**	0.864	**2.189**	2.555
3	**0.444**	0.440	0.147	0.147

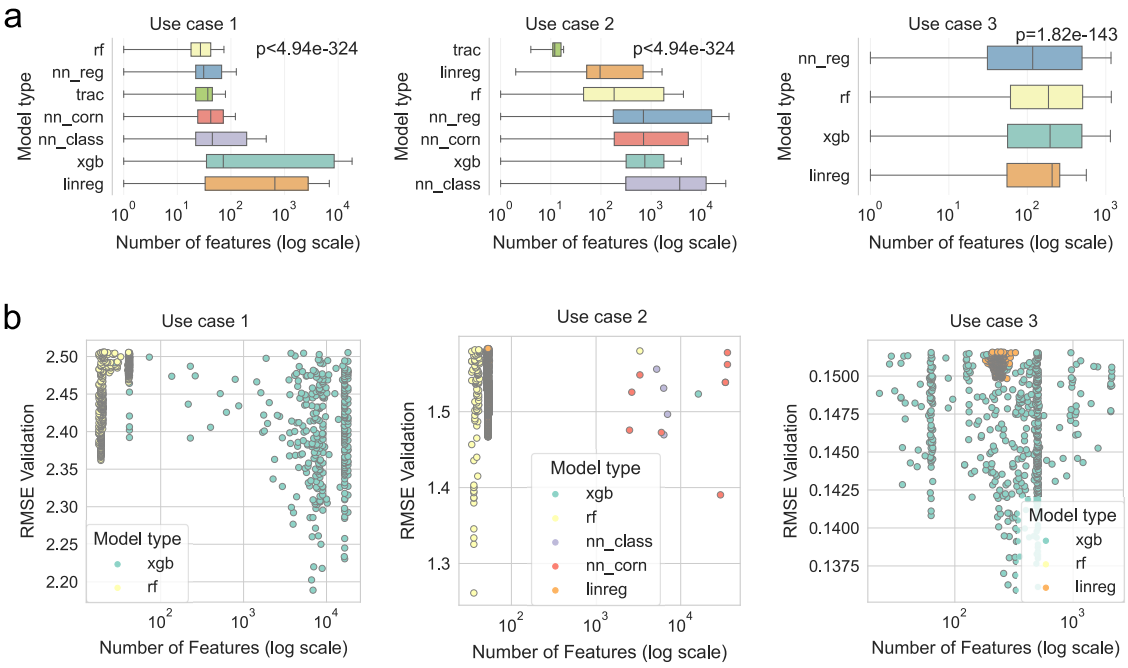

Figure 5: Relationship between the number of selected features and validation root mean squared error (RMSE) for all three use cases (**a**) grouped by model type and (**b**) for the top 1'000 best-performing configurations. P-values report Kruskal–Wallis tests across groups.

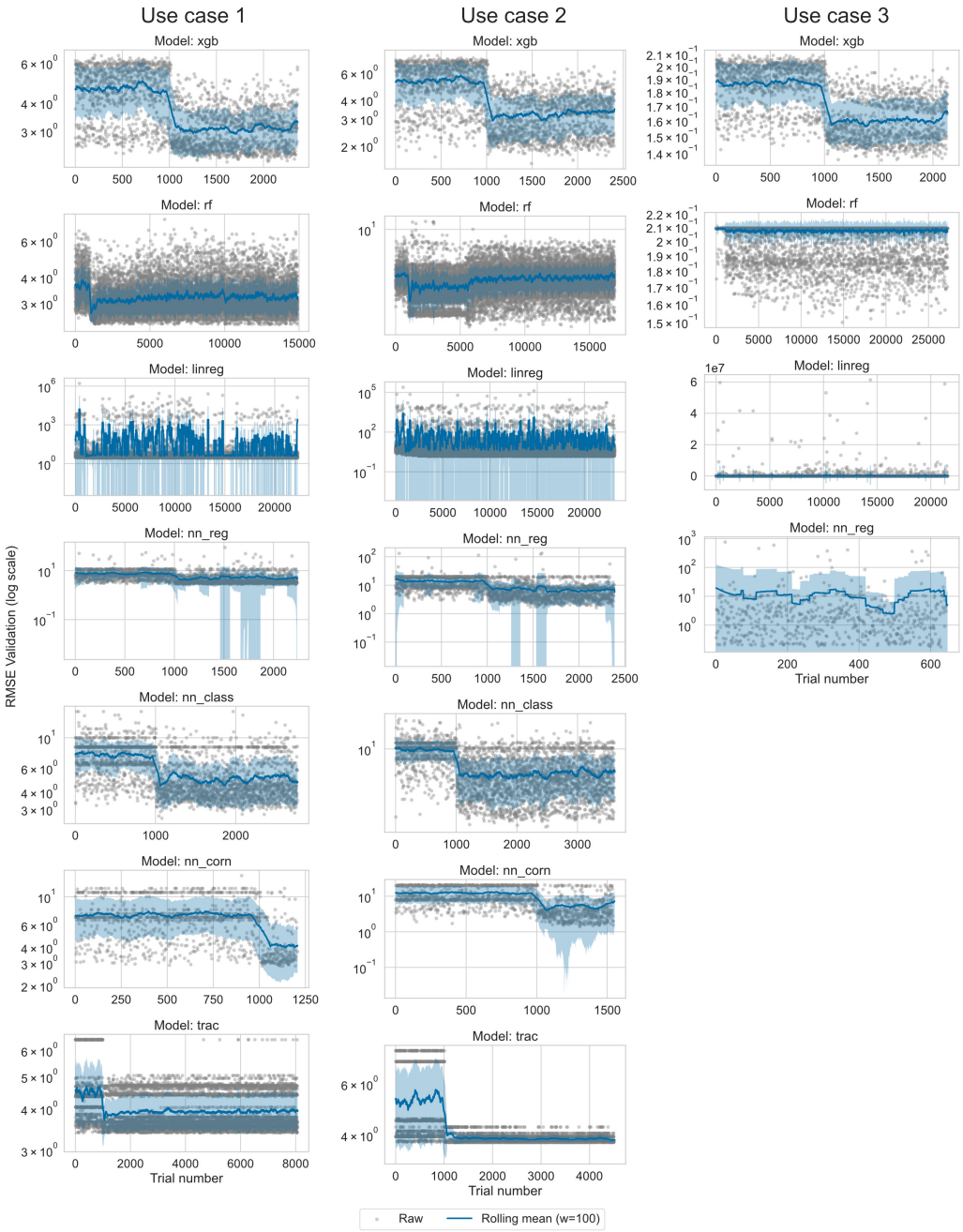

Figure 6: Validation root mean squared error (RMSE) performance trajectories across trials for all three use cases. Dots represent the raw trials, blue lines represent the rolling mean trajectories calculated with a window size of 100 trials and shaded areas represent the rolling standard deviations.

4 Discussion

ritme provides a domain-adapted framework for optimized predictive modelling on microbial sequencing data that addresses key statistical properties of these datasets — sparsity, high dimensionality, compositionality, and hierarchical structure — while jointly optimizing feature representation and model class.

Across three independent use cases, *ritme* outperformed original study modelling pipelines (Tables 5 to 7) and a generic *auto-sklearn* baseline (Table 8). The comparison with *auto-sklearn* emphasizes that domain knowledge matters in terms of predictive performance, with *ritme*'s microbiome-specific feature engineering options outperforming generic search spaces for tabular data. Beyond predictive performance, *ritme* gives users insights into how feature and model choices shape predictive performance (Figures 4 to 6). In practice, users can also inject their own expertise by constraining the search to biologically plausible subspaces or by seeding initial configurations (Sections 2.3 and 2.4). Moreover, while *ritme* is optimized for microbiome sequencing data, it can be applied to any 2D multi-omics feature count table by deselecting non-applicable feature engineering steps (e.g., taxonomic aggregation or compositional transforms).

As an open-source software built in a modular fashion, we plan different future extensions to *ritme*. While *ritme* includes model inference based on regression, classification, and ordinal regression, its present evaluation is focused on regression tasks. Future extensions to classification tasks are warranted. Moreover, automatically selecting and adapting the hyperparameter search algorithm to the given data's search space characteristics would enable even more data-driven inference of the optimal feature-model configuration. Also, instead of optimizing feature-model combinations per model algorithm, a joint opimization should be added. Finally, approaches to identify minimal, stable feature sets that are reliably predictive could be integrated to support parsimonious biomarker discovery.

In summary, *ritme* offers a data-driven, compute-efficient framework that standardizes the joint optimization of feature representations and models for microbiome sequencing data. By combining domain-specific feature engineering with optimized hyperparameter search and transparent experiment tracking, it delivers state-of-the-art predictive performance while providing actionable insight into which biological and statistical choices drive that performance. By replacing subjective, *ad hoc* modelling decisions with systematic optimization, *ritme* enables researchers to identify best-performing feature-model combinations from microbiome sequencing data in a standardized manner. We anticipate that *ritme* will help mitigate reproducibility

concerns in microbiome research [73] and accelerate robust, generalizable discovery.

Key Points

- *ritme* provides automated machine learning tailored to microbiome sequencing data, systematically optimizing both feature representations and model algorithms while delivering insights into the drivers of predictive performance.
- *ritme* implements microbiome-specific feature engineering — including taxonomic aggregation, sparsity-aware selection, compositional transforms, and metadata enrichment — addressing the high-dimensional, sparse, compositional, and hierarchical nature of microbiome sequencing data.
- The open-source package features state-of-the-art hyperparameter optimization with distributed computing and modular architecture, enabling efficient exploration of vast search spaces and seamless extensibility to other multi-omics modalities.

Acknowledgements

This work was supported by an ETH Zurich Doc.Mobility Fellowship awarded to Anja Adamov. We acknowledge the use of GitHub Copilot to assist with software testing, debugging, documentation, data analysis, and manuscript editing.

Data Availability

Reproducible examples on using *ritme* through the CLI or Python API and a templates for evaluating *ritme* models with MLflow or Weights and Biases are available in its GitHub repository at `https://github.com/adamovanja/ritme`, archived on Zenodo at `https://zenodo.org/records/17537117`. All real-world use case examples presented in this article, including data fetching, processing and modelling, are available under `https://github.com/bokulich-publications/ritme_examples`.

Author Contributions

AA: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing; CLM: Conceptualization, Methodology, Supervision, Writing – review & editing; NAB: Conceptualization, Methodology, Funding acquisition, Resources, Supervision, Writing – review & editing.

References

[1] Jason Lloyd-Price, Cesar Arze, Ashwin N. Ananthakrishnan, Melanie Schirmer, Julian Avila-Pacheco, Tiffany W. Poon, Elizabeth Andrews, Nadim J. Ajami, Kevin S. Bonham, Colin J. Brislawn, David Casero, Holly Courtney, Antonio Gonzalez, Thomas G. Graeber, A. Brantley Hall, Kathleen Lake, Carol J. Landers, Himel Mallick, Damian R. Plichta, Mahadev Prasad, Gholamali Rahnavard, Jenny Sauk, Dmitry Shungin, Yoshiki Vázquez-Baeza, Richard A. White, Jonathan Braun, Lee A. Denson, Janet K. Jansson, Rob Knight, Subra Kugathasan, Dermot P. B. McGovern, Joseph F. Petrosino, Thaddeus S. Stappenbeck, Harland S. Winter, Clary B. Clish, Eric A. Franzosa, Hera Vlamakis, Ramnik J. Xavier, and Curtis Huttenhower. Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. *Nature*, 569(7758): 655–662, 2019. ISSN 0028-0836. doi: 10.1038/s41586-019-1237-9. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6650278/`.

[2] Maayan Levy, Aleksandra A. Kolodziejczyk, Christoph A. Thaiss, and Eran Elinav. Dysbiosis and the immune system. *Nature Reviews Immunology*, 17(4):219–232, April 2017. ISSN 1474-1741. doi: 10.1038/nri.2017.7. URL `https://www.nature.com/articles/nri.2017.7`.

[3] Peter J. Turnbaugh, Ruth E. Ley, Micah Hamady, Claire M. Fraser-Liggett, Rob Knight, and Jeffrey I. Gordon. The Human Microbiome Project. *Nature*, 449(7164):804–810, October 2007. ISSN 1476-4687. doi: 10.1038/nature06244. URL `https://www.nature.com/articles/nature06244`.

[4] Martin Hartmann and Johan Six. Soil structure and microbiome functions in agroecosystems. *Nature Reviews Earth & Environment*, 4(1):4–18, January 2023. ISSN 2662-138X. doi: 10.1038/s43017-022-00366-w. URL `https://www.nature.com/articles/s43017-022-00366-w`.

[5] Nicholas A. Bokulich, Thomas S. Collins, Chad Masarweh, Greg Allen, Hildegarde Heymann, Susan E. Ebeler, and David A. Mills. Associations among Wine Grape Microbiome, Metabolome, and Fermentation Behavior Suggest Microbial Contribution to Regional Wine Characteristics. *mBio*, 7(3):10.1128/mbio.00631–16, June 2016. doi: 10.1128/mbio.00631-16. URL `https://journals.asm.org/doi/10.1128/mbio.00631-16`.

[6] Ian Kaplan, Nicholas A. Bokulich, J. Gregory Caporaso, Laramy S. Enders, Wadih Ghanem, and Kathryn S. Ingerslew. Phylogenetic farming:

Can evolutionary history predict crop rotation via the soil microbiome? *Evolutionary Applications*, 13(8):1984–1999, 2020. ISSN 1752-4571. doi: 10.1111/eva.12956. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/eva.12956.

[7] Noah W. Sokol, Eric Slessarev, Gianna L. Marschmann, Alexa Nicolas, Steven J. Blazewicz, Eoin L. Brodie, Mary K. Firestone, Megan M. Foley, Rachel Hestrin, Bruce A. Hungate, Benjamin J. Koch, Bram W. Stone, Matthew B. Sullivan, Olivier Zablocki, and Jennifer Pett-Ridge. Life and death in the soil microbiome: How ecological processes influence biogeochemistry. 20(7):415–430. ISSN 1740-1534. doi: 10.1038/s41579-022-00695-z. URL https://www.nature.com/articles/s41579-022-00695-z.

[8] Shinichi Sunagawa, Luis Pedro Coelho, Samuel Chaffron, Jens Roat Kultima, Karine Labadie, Guillem Salazar, Bardya Djahanschiri, Georg Zeller, Daniel R. Mende, Adriana Alberti, Francisco M. Cornejo-Castillo, Paul I. Costea, Corinne Cruaud, Francesco d'Ovidio, Stefan Engelen, Isabel Ferrera, Josep M. Gasol, Lionel Guidi, Falk Hildebrand, Florian Kokoszka, Cyrille Lepoivre, Gipsi Lima-Mendez, Julie Poulain, Bonnie T. Poulos, Marta Royo-Llonch, Hugo Sarmento, Sara Vieira-Silva, Céline Dimier, Marc Picheral, Sarah Searson, Stefanie Kandels-Lewis, Tara Oceans coordinators, Chris Bowler, Colomban de Vargas, Gabriel Gorsky, Nigel Grimsley, Pascal Hingamp, Daniele Iudicone, Olivier Jaillon, Fabrice Not, Hiroyuki Ogata, Stephane Pesant, Sabrina Speich, Lars Stemmann, Matthew B. Sullivan, Jean Weissenbach, Patrick Wincker, Eric Karsenti, Jeroen Raes, Silvia G. Acinas, and Peer Bork. Structure and function of the global ocean microbiome. *Science*, 348(6237):1261359, May 2015. doi: 10.1126/science.1261359. URL https://www.science.org/doi/10.1126/science.1261359.

[9] Luke R. Thompson, Jon G. Sanders, Daniel McDonald, Amnon Amir, Joshua Ladau, Kenneth J. Locey, Robert J. Prill, Anupriya Tripathi, Sean M. Gibbons, Gail Ackermann, Jose A. Navas-Molina, Stefan Janssen, Evguenia Kopylova, Yoshiki Vázquez-Baeza, Antonio González, James T. Morton, Siavash Mirarab, Zhenjiang Zech Xu, Lingjing Jiang, Mohamed F. Haroon, Jad Kanbar, Qiyun Zhu, Se Jin Song, Tomasz Kosciolek, Nicholas A. Bokulich, Joshua Lefler, Colin J. Brislawn, Gregory Humphrey, Sarah M. Owens, Jarrad Hampton-Marcell, Donna Berg-Lyons, Valerie McKenzie, Noah Fierer, Jed A. Fuhrman, Aaron Clauset, Rick L. Stevens, Ashley Shade, Katherine S. Pollard, Kelly D. Goodwin, Janet K. Jansson, Jack A. Gilbert, and

Rob Knight. A communal catalogue reveals Earth's multiscale microbial diversity. *Nature*, 551(7681):457–463, November 2017. ISSN 1476-4687. doi: 10.1038/nature24621. URL https://www.nature.com/articles/nature24621.

[10] Dan Knights, Justin Kuczynski, Omry Koren, Ruth E Ley, Dawn Field, Rob Knight, Todd Z DeSantis, and Scott T Kelley. Supervised classification of microbiota mitigates mislabeling errors. *The ISME Journal*, 5 (4):570–573, April 2011. ISSN 1751-7362. doi: 10.1038/ismej.2010.148. URL https://doi.org/10.1038/ismej.2010.148.

[11] Nicholas A. Bokulich, Matthew R. Dillon, Evan Bolyen, Benjamin D. Kaehler, Gavin A. Huttley, and J. Gregory Caporaso. Q2-sample-classifier: Machine-learning tools for microbiome classification and regression. *Journal of Open Source Software*, 3(30):934, October 2018. ISSN 2475-9066. doi: 10.21105/joss.00934. URL https://joss.theoj.org/papers/10.21105/joss.00934.

[12] Alan Le Goallec, Braden T. Tierney, Jacob M. Luber, Evan M. Cofer, Aleksandar D. Kostic, and Chirag J. Patel. A systematic machine learning and data type comparison yields metagenomic predictors of infant age, sex, breastfeeding, antibiotic usage, country of origin, and delivery type. *PLOS Computational Biology*, 16(5):e1007895, May 2020. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007895. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007895.

[13] Edoardo Pasolli, Duy Tin Truong, Faizan Malik, Levi Waldron, and Nicola Segata. Machine Learning Meta-analysis of Large Metagenomic Datasets: Tools and Biological Insights. *PLOS Computational Biology*, 12(7):e1004977, July 2016. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004977. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004977.

[14] Yi-Hui Zhou and Paul Gallins. A Review and Tutorial of Machine Learning Methods for Microbiome Host Trait Prediction. *Frontiers in Genetics*, 10, June 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.00579. URL https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2019.00579/full.

[15] Jakob Wirbel, Konrad Zych, Morgan Essex, Nicolai Karcher, Ece Kartal, Guillem Salazar, Peer Bork, Shinichi Sunagawa, and Georg Zeller. Microbiome meta-analysis and cross-disease comparison enabled by the

SIAMCAT machine learning toolbox. *Genome Biology*, 22(1):93, March 2021. ISSN 1474-760X. doi: 10.1186/s13059-021-02306-1. URL https://doi.org/10.1186/s13059-021-02306-1.

[16] Isabel Moreno-Indias, Leo Lahti, Miroslava Nedyalkova, Ilze Elbere, Gennady Roshchupkin, Muhamed Adilovic, Onder Aydemir, Burcu Bakir-Gungor, Enrique Carrillo-de Santa Pau, Domenica D'Elia, Mahesh S. Desai, Laurent Falquet, Aycan Gundogdu, Karel Hron, Thomas Klammsteiner, Marta B. Lopes, Laura Judith Marcos-Zambrano, Cláudia Marques, Michael Mason, Patrick May, Lejla Pašić, Gianvito Pio, Sándor Pongor, Vasilis J. Promponas, Piotr Przymus, Julio Saez-Rodriguez, Alexia Sampri, Rajesh Shigdel, Blaz Stres, Ramona Suharoschi, Jaak Truu, Ciprian-Octavian Truică, Baiba Vilne, Dimitrios Vlachakis, Ercument Yilmaz, Georg Zeller, Aldert L. Zomer, David Gómez-Cabrero, and Marcus J. Claesson. Statistical and Machine Learning Techniques in Human Microbiome Studies: Contemporary Challenges and Solutions. *Frontiers in Microbiology*, 12, February 2021. ISSN 1664-302X. doi: 10.3389/fmicb.2021.635781. URL https://www.frontiersin.org/journals/microbiology/articles/10.3389/fmicb.2021.635781/full.

[17] Sarah L. Westcott and Patrick D. Schloss. De novo clustering methods outperform reference-based methods for assigning 16S rRNA gene sequences to operational taxonomic units. *PeerJ*, 3:e1487, December 2015. ISSN 2167-8359. doi: 10.7717/peerj.1487. URL https://peerj.com/articles/1487.

[18] Evguenia Kopylova, Jose A. Navas-Molina, Céline Mercier, Zhenjiang Zech Xu, Frédéric Mahé, Yan He, Hong-Wei Zhou, Torbjørn Rognes, J. Gregory Caporaso, and Rob Knight. Open-Source Sequence Clustering Methods Improve the State Of the Art. *mSystems*, 1(1): 10.1128/msystems.00003–15, February 2016. doi: 10.1128/msystems.00003-15. URL https://journals.asm.org/doi/10.1128/msystems.00003-15.

[19] Shinichi Sunagawa, Daniel R. Mende, Georg Zeller, Fernando Izquierdo-Carrasco, Simon A. Berger, Jens Roat Kultima, Luis Pedro Coelho, Manimozhiyan Arumugam, Julien Tap, Henrik Bjørn Nielsen, Simon Rasmussen, Søren Brunak, Oluf Pedersen, Francisco Guarner, Willem M. de Vos, Jun Wang, Junhua Li, Joël Doré, S. Dusko Ehrlich, Alexandros Stamatakis, and Peer Bork. Metagenomic species profiling using universal phylogenetic marker genes. *Nature Methods*, 10(12):

1196–1199, December 2013. ISSN 1548-7105. doi: 10.1038/nmeth.2693. URL https://www.nature.com/articles/nmeth.2693.

[20] Alessio Milanese, Daniel R. Mende, Lucas Paoli, Guillem Salazar, Hans-Joachim Ruscheweyh, Miguelangel Cuenca, Pascal Hingamp, Renato Alves, Paul I. Costea, Luis Pedro Coelho, Thomas S. B. Schmidt, Alexandre Almeida, Alex L. Mitchell, Robert D. Finn, Jaime Huerta-Cepas, Peer Bork, Georg Zeller, and Shinichi Sunagawa. Microbial abundance, activity and population genomic profiling with mOTUs2. *Nature Communications*, 10(1):1014, March 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-08844-4. URL https://www.nature.com/articles/s41467-019-08844-4.

[21] Benjamin J Callahan, Paul J McMurdie, and Susan P Holmes. Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *The ISME Journal*, 11(12):2639–2643, December 2017. ISSN 1751-7362. doi: 10.1038/ismej.2017.119. URL https://doi.org/10.1038/ismej.2017.119.

[22] Gregory B. Gloor, Jean M. Macklaim, Vera Pawlowsky-Glahn, and Juan J. Egozcue. Microbiome Datasets Are Compositional: And This Is Not Optional. *Frontiers in Microbiology*, 8, 2017. ISSN 1664-302X. doi: 10.3389/fmicb.2017.02224. URL https://www.frontiersin.org/articles/10.3389/fmicb.2017.02224/full.

[23] Georgios Papoutsoglou, Sonia Tarazona, Marta B. Lopes, Thomas Klammsteiner, Eliana Ibrahimi, Julia Eckenberger, Pierfrancesco Novielli, Alberto Tonda, Andrea Simeon, Rajesh Shigdel, Stéphane Béreux, Giacomo Vitali, Sabina Tangaro, Leo Lahti, Andriy Temko, Marcus J. Claesson, and Magali Berland. Machine learning approaches in microbiome research: Challenges and best practices. *Frontiers in Microbiology*, 14, September 2023. ISSN 1664-302X. doi: 10.3389/fmicb.2023.1261889. URL https://www.frontiersin.org/journals/microbiology/articles/10.3389/fmicb.2023.1261889/full.

[24] Shimeng Huang, Elisabeth Ailer, Niki Kilbertus, and Niklas Pfister. Supervised learning and model analysis with compositional data. *PLOS Computational Biology*, 19(6):e1011240, June 2023. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1011240. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011240.

[25] Randal S. Olson and Jason H. Moore. TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning. In Frank Hut-

ter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Automated Machine Learning: Methods, Systems, Challenges*, pages 151–160. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05318-5. doi: 10.1007/978-3-030-05318-5_8. URL https://doi.org/10.1007/978-3-030-05318-5_8.

[26] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05317-8 978-3-030-05318-5. doi: 10.1007/978-3-030-05318-5. URL http://link.springer.com/10.1007/978-3-030-05318-5.

[27] Ioannis Tsamardinos, Paulos Charonyktakis, Georgios Papoutsoglou, Giorgos Borboudakis, Kleanthi Lakiotaki, Jean Claude Zenklusen, Hartmut Juhl, Ekaterini Chatzaki, and Vincenzo Lagani. Just Add Data: Automated predictive modeling for knowledge discovery and feature selection. *npj Precision Oncology*, 6(1):38, June 2022. ISSN 2397-768X. doi: 10.1038/s41698-022-00274-8. URL https://www.nature.com/articles/s41698-022-00274-8.

[28] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and Robust Automated Machine Learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://papers.nips.cc/paper_files/paper/2015/hash/11d0e6287202fced83f79975ec59a3a6-Abstract.html.

[29] Laura Judith Marcos-Zambrano, Kanita Karaduzovic-Hadziabdic, Tatjana Loncar Turukalo, Piotr Przymus, Vladimir Trajkovik, Oliver Aasmets, Magali Berland, Aleksandra Gruca, Jasminka Hasic, Karel Hron, Thomas Klammsteiner, Mikhail Kolev, Leo Lahti, Marta B. Lopes, Victor Moreno, Irina Naskinova, Elin Org, Inês Paciência, Georgios Papoutsoglou, Rajesh Shigdel, Blaz Stres, Baiba Vilne, Malik Yousef, Eftim Zdravevski, Ioannis Tsamardinos, Enrique Carrillo de Santa Pau, Marcus J. Claesson, Isabel Moreno-Indias, and Jaak Truu. Applications of Machine Learning in Human Microbiome Studies: A Review on Feature Selection, Biomarker Identification, Disease Prediction and Treatment. *Frontiers in Microbiology*, 12, February 2021. ISSN 1664-302X. doi: 10.3389/fmicb.2021.634511. URL https://www.frontiersin.org/journals/microbiology/articles/10.3389/fmicb.2021.634511/full.

[30] John P. A. Ioannidis. Why Most Discovered True Associations Are Inflated. *Epidemiology*, 19(5):640, September 2008. ISSN 1044-3983. doi: 10.1097/EDE.0b013e31818131e7. URL https://journals.lww.com/epidem/abstract/2008/09000/why_most_discovered_true_associations_are_inflated.2.aspx.

[31] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A Research Platform for Distributed Model Selection and Training, July 2018. URL http://arxiv.org/abs/1807.05118.

[32] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A System for Massively Parallel Hyperparameter Tuning, March 2020. URL http://arxiv.org/abs/1810.05934.

[33] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization, June 2018. URL http://arxiv.org/abs/1603.06560.

[34] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA, July 2019. Association for Computing Machinery. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330701. URL https://dl.acm.org/doi/10.1145/3292500.3330701.

[35] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html.

[36] Shuhei Watanabe. Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance. URL http://arxiv.org/abs/2304.11127.

[37] Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*,

11(1):1–18, March 2003. ISSN 1063-6560. doi: 10.1162/106365603321828970. URL https://ieeexplore.ieee.org/document/6790790.

[38] Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9 (2):159–195, June 2001. ISSN 1063-6560, 1530-9304. doi: 10.1162/106365601750190398. URL https://direct.mit.edu/evco/article/9/2/159-195/892.

[39] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2. URL https://www.nature.com/articles/s41592-019-0686-2.

[40] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. Hong, A. Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, Fen Xie, and Corey Zumar. Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.*, 2018. URL https://www.semanticscholar.org/paper/Accelerating-the-Machine-Learning-Lifecycle-with-Zaharia-Chen/b2e0b79e6f180af2e0e559f2b1faba66b2bd578a.

[41] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/.

[42] J. Aitchison. The Statistical Analysis of Compositional Data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, January 1982. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1982.tb01195.x. URL https://doi.org/10.1111/j.2517-6161.1982.tb01195.x.

[43] J. J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, and C. Barceló-Vidal. Isometric Logratio Transformations for Compositional Data Analysis. *Mathematical Geology*, 35(3):279–300, April 2003. ISSN 1573-8868. doi: 10.1023/A:1023818214614. URL https://doi.org/10.1023/A:1023818214614.

[44] Jai Ram Rideout, Greg Caporaso, Evan Bolyen, Daniel McDonald, Yoshiki Vázquez Baeza, Jorge Cañardo Alastuey, Anders Pitman, Jamie Morton, Jose Navas, Kestrel Gorlick, Justine Debelius, Zech Xu, llcooljohn, adamrp, Joshua Shorenstein, Laurent Luce, Will Van Treuren, charudatta-navare, Antonio Gonzalez, Colin J. Brislawn, Weronika Patena, Karen Schwarzberg, teravest, Jens Reeder, shiffer1, Igor Sfiligoi, nbresnick, Qiyun Zhu, Dr K. D. Murray, and Karan Sharma. Biocore/scikit-bio: Scikit-bio 0.5.9: Maintenance release. Zenodo, August 2023. URL https://zenodo.org/records/8209901.

[45] Daniel McDonald, Jose C Clemente, Justin Kuczynski, Jai Ram Rideout, Jesse Stombaugh, Doug Wendel, Andreas Wilke, Susan Huse, John Hufnagle, Folker Meyer, Rob Knight, and J Gregory Caporaso. The Biological Observation Matrix (BIOM) format or: How I learned to stop worrying and love the ome-ome. *GigaScience*, 1(1):2047–217X–1–7, December 2012. ISSN 2047-217X. doi: 10.1186/2047-217X-1-7. URL https://doi.org/10.1186/2047-217X-1-7.

[46] The pandas development team. Pandas-dev/pandas: Pandas. Zenodo, April 2024. URL https://zenodo.org/records/10957263.

[47] Wes McKinney. Data Structures for Statistical Computing in Python. *scipy*, May 2010. doi: 10.25080/Majora-92bf1922-00a. URL https://proceedings.scipy.org/articles/Majora-92bf1922-00a.

[48] Claude Shannon and Warren Weaver. The Mathematical Theory of Communication.

[49] Xintong Shi, Wenzhi Cao, and Sebastian Raschka. Deep Neural Networks for Rank-Consistent Ordinal Regression Based On Conditional Probabilities. *Pattern Analysis and Applications*, 26(3):941–955, August 2023. ISSN 1433-7541, 1433-755X. doi: 10.1007/s10044-023-01181-9. URL http://arxiv.org/abs/2111.08851.

[50] Jacob Bien, Xiaohan Yan, Léo Simpson, and Christian L. Müller. Tree-aggregated predictive modeling of microbiome data. *Scientific Reports*, 11(1):14505, July 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-93645-3. URL https://www.nature.com/articles/s41598-021-93645-3.

[51] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, March 2015. URL http://arxiv.org/abs/1502.03167.

[52] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. 12(85): 2825–2830. ISSN 1533-7928. URL http://jmlr.org/papers/v12/pedregosa11a.html.

[53] Léo Simpson, Patrick L. Combettes, and Christian L. Müller. C-lasso - a Python package for constrained sparse and robust regression and classification. *Journal of Open Source Software*, 6(57):2844, January 2021. ISSN 2475-9066. doi: 10.21105/joss.02844. URL https://joss.theoj.org/papers/10.21105/joss.02844.

[54] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. doi: 10.1145/2939672.2939785. URL http://arxiv.org/abs/1603.02754.

[55] William Falcon and The PyTorch Lightning team. PyTorch lightning, March 2019. URL https://github.com/Lightning-AI/lightning.

[56] Sebastián Ramírez. Typer. URL https://github.com/fastapi/typer.

[57] Andrew Hunt and David Thomas. *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley Professional, Boston, 2011. ISBN 978-0-201-61622-4.

[58] Nicholas A. Bokulich, Michal Ziemski, Michael S. Robeson, and Benjamin D. Kaehler. Measuring the microbiome: Best practices for developing and benchmarking microbiomics methods. *Computational and Structural Biotechnology Journal*, 18:4048–4062, January 2020. ISSN 2001-0370. doi: 10.1016/j.csbj.2020.11.049. URL https://www.sciencedirect.com/science/article/pii/S200103702030516X.

[59] Sathish Subramanian, Sayeeda Huq, Tanya Yatsunenko, Rashidul Haque, Mustafa Mahfuz, Mohammed A. Alam, Amber Benezra, Joseph DeStefano, Martin F. Meier, Brian D. Muegge, Michael J. Barratt, Laura G. VanArendonk, Qunyuan Zhang, Michael A. Province, William A. Petri Jr, Tahmeed Ahmed, and Jeffrey I. Gordon. Persistent gut microbiota immaturity in malnourished Bangladeshi children. *Nature*, 510(7505):417–421, June 2014. ISSN 1476-4687. doi: 10.1038/nature13421. URL https://www.nature.com/articles/nature13421.

[60] Ramiro Logares, Shinichi Sunagawa, Guillem Salazar, Francisco M. Cornejo-Castillo, Isabel Ferrera, Hugo Sarmento, Pascal Hingamp, Hiroyuki Ogata, Colomban de Vargas, Gipsi Lima-Mendez, Jeroen Raes, Julie Poulain, Olivier Jaillon, Patrick Wincker, Stefanie Kandels-Lewis, Eric Karsenti, Peer Bork, and Silvia G. Acinas. Metagenomic 16S rDNA Illumina tags are a powerful alternative to amplicon sequencing to explore diversity and structure of microbial communities. *Environmental Microbiology*, 16(9):2659–2671, 2014. ISSN 1462-2920. doi: 10.1111/1462-2920.12250. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/1462-2920.12250`.

[61] Suguru Nishijima, Evelina Stankevic, Oliver Aasmets, Thomas S. B. Schmidt, Naoyoshi Nagata, Marisa Isabell Keller, Pamela Ferretti, Helene Bæk Juel, Anthony Fullam, Shahriyar Mahdi Robbani, Christian Schudoma, Johanne Kragh Hansen, Louise Aas Holm, Mads Israelsen, Robert Schierwagen, Nikolaj Torp, Anja Telzerow, Rajna Hercog, Stefanie Kandels, Diënty H. M. Hazenbrink, Manimozhiyan Arumugam, Flemming Bendtsen, Charlotte Brøns, Cilius Esmann Fonvig, Jens-Christian Holm, Trine Nielsen, Julie Steen Pedersen, Maja Sofie Thiele, Jonel Trebicka, Elin Org, Aleksander Krag, Torben Hansen, Michael Kuhn, Peer Bork, Torben Hansen, Matthias Mann, Jelle Matthijnssens, Aleksander Krag, Peer Bork, Manimozhiyan Arumugam, Jonel Trebicka, Morten Karsdal, Ema Anastasiadou, Hans Israelsen, Hans Olav Melberg, Cristina Legido-Quigley, and Maja Thiele. Fecal microbial load is a major determinant of gut microbiome variation and a confounder for disease associations. *Cell*, 188(1):222–236.e15, January 2025. ISSN 0092-8674. doi: 10.1016/j.cell.2024.10.022. URL `https://www.sciencedirect.com/science/article/pii/S0092867424012042`.

[62] Isabelle Guyon, Lisheng Sun-Hosoya, Marc Boullé, Hugo Jair Escalante, Sergio Escalera, Zhengying Liu, Damir Jajetic, Bisakha Ray, Mehreen Saeed, Michèle Sebag, Alexander Statnikov, Wei-Wei Tu, and Evelyne Viegas. Analysis of the AutoML Challenge Series 2015–2018. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Automated Machine Learning: Methods, Systems, Challenges*, pages 177–219. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05318-5. doi: 10.1007/978-3-030-05318-5_10. URL `https://doi.org/10.1007/978-3-030-05318-5_10`.

[63] Michal Ziemski, Anja Adamov, Lina Kim, Lena Flörl, and Nicholas A Bokulich. Reproducible acquisition, management and meta-analysis of nucleotide sequence (meta)data using q2-fondue. *Bioinformatics*, 38

(22):5081–5091, November 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac639. URL https://doi.org/10.1093/bioinformatics/btac639.

[64] Yuichi Kodama, Martin Shumway, Rasko Leinonen, and on behalf of the International Nucleotide Sequence Database Collaboration. The sequence read archive: Explosive growth of sequencing data. *Nucleic Acids Research*, 40(D1):D54–D56, January 2012. ISSN 0305-1048. doi: 10.1093/nar/gkr854. URL https://doi.org/10.1093/nar/gkr854.

[65] Marcel Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17(1):10–12, May 2011. ISSN 2226-6089. doi: 10.14806/ej.17.1.200. URL https://journal.embnet.org/index.php/embnetjournal/article/view/200.

[66] Daniel McDonald, Morgan N. Price, Julia Goodrich, Eric P. Nawrocki, Todd Z. DeSantis, Alexander Probst, Gary L. Andersen, Rob Knight, and Philip Hugenholtz. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *The ISME journal*, 6(3):610–618, March 2012. ISSN 1751-7370. doi: 10.1038/ismej.2011.139.

[67] Torbjørn Rognes, Tomáš Flouri, Ben Nichols, Christopher Quince, and Frédéric Mahé. VSEARCH: A versatile open source tool for metagenomics. 4:e2584. ISSN 2167-8359. doi: 10.7717/peerj.2584. URL https://peerj.com/articles/2584.

[68] Nicholas A. Bokulich, Benjamin D. Kaehler, Jai Ram Rideout, Matthew Dillon, Evan Bolyen, Rob Knight, Gavin A. Huttley, and J. Gregory Caporaso. Optimizing taxonomic classification of marker-gene amplicon sequences with QIIME 2's q2-feature-classifier plugin. *Microbiome*, 6 (1):90, May 2018. ISSN 2049-2618. doi: 10.1186/s40168-018-0470-z. URL https://doi.org/10.1186/s40168-018-0470-z.

[69] Kazutaka Katoh and Daron M. Standley. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution*, 30(4):772–780, April 2013. ISSN 0737-4038. doi: 10.1093/molbev/mst010. URL https://doi.org/10.1093/molbev/mst010.

[70] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE*, 5(3):e9490, March 2010. ISSN 1932-6203. doi: 10.1371/journal.

pone.0009490. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2835736/.

[71] Christian Quast, Elmar Pruesse, Pelin Yilmaz, Jan Gerken, Timmy Schweer, Pablo Yarza, Jörg Peplies, and Frank Oliver Glöckner. The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. *Nucleic Acids Research*, 41(D1):D590–D596, January 2013. ISSN 0305-1048. doi: 10.1093/nar/gks1219. URL https://doi.org/10.1093/nar/gks1219.

[72] Maxwell A. Konnaris, Manan Saxena, Nicole Lazar, and Justin D. Silverman. Uncertainty Modeling Outperforms Machine Learning for Microbiome Data Analysis, September 2025. ISSN 2692-8205. URL https://www.biorxiv.org/content/10.1101/2025.09.12.675956v1.

[73] Patrick D. Schloss. Identifying and Overcoming Threats to Reproducibility, Replicability, Robustness, and Generalizability in Microbiome Research. *mBio*, 9(3):10.1128/mbio.00525–18, June 2018. doi: 10.1128/mbio.00525-18. URL https://journals.asm.org/doi/10.1128/mbio.00525-18.

Supplementary data

Table S1: Hyperparameters and their default evaluated ranges for each predictive model in *ritme*. Categorical hyperparameters list discrete evaluated values; continuous hyperparameters show the sampled interval $[\min, \max]$ with log indicating log-uniform sampling and int indicating that only integer values are considered.

Model	Categorical	Continuous
`linreg`	–	`alpha`: $[10^{-5}, 10^{2}]$ (log) `l1_ratio`: $[0, 1]$
`rf`	`max_depth`: {2, 4, 8, 16, 32, None} `max_features`: {None, "sqrt", "log2" 0.1, 0.2, 0.3, 0.5} `bootstrap`: {True, False}	`n_estimators`: $[20, 3000]$ (int) `min_samples_split`: $[10^{-3}, 10^{-1}]$ (log) `min_samples_leaf`: $[10^{-3}, 10^{-1}]$ (log) `min_weight_fraction_leaf`: $[0, 10^{-2}]$ `min_impurity_decrease`: $[0, 0.5]$
`trac`	–	`lambda`: $[10^{-3}, 1]$ (log)
`xgb`	–	`n_estimators`: $[20, 3000]$ (int) `max_depth`: $[2, 10]$ (int) `min_child_weight`: $[0, 4]$ (int) `subsample`: $[0.7, 1.0]$ `eta`: $[10^{-2}, 0.3]$ (log) `num_parallel_tree`: $[1, 4]$ (int) `gamma`: $[10^{-3}, 5]$ (log) `reg_alpha`: $[10^{-10}, 1]$ (log) `reg_lambda`: $[10^{-10}, 1]$ (log) `colsample_bytree`: $[0.3, 1.0]$
`nn_reg`, `nn_class`, `nn_corn`	`n_units_hl`x: {32, 64, 128, 256, 512} `learning_rate`: {1e-2, 5e-3, 1e-3 5e-4, 1e-4, 5e-5, 1e-5} `batch_size`: {32, 64, 128, 256} `epochs`: {10, 50, 100, 200}	`n_hidden_layers`: $[1, 30]$ (int) `dropout_rate`: $[0, 0.5]$ `weight_decay`: $[10^{-6}, 10^{-2}]$ (log) `early_stopping_patience`: $[2, 30]$ (int) `early_stopping_min_delta`: $[10^{-5}, 10^{-2}]$ (log)

a

```python
# Infer best feature-model configuration
training_set, test_set = split_train_test(
    metadata, feature_table
)
best_models, path_to_exp = find_best_model_config(
    experiment_config, training_set, taxonomy, phylogeny
)
best_linreg_model = best_models["linreg"]
type(best_linreg_model)
```

```
> ritme.evaluate_models.TunedModel
```

b

```python
# Perform predictions
test_pred = best_linreg_model.predict(test_set, "test")
```

```
> array([ 3.93, 15.06, 6. , 11.14, 1.28, 18.46, 18.11])
```

c

```python
# Extract best feature engineering
best_linreg_model.data_config
```

```
> {'data_aggregation': 'tax_order',
 'data_selection': 'abundance_threshold',
 'data_selection_t': 2.297646358449064e-08,
 'data_transform': 'rank',
 'data_enrich': None,
 'data_enrich_with': None,
 'data_alr_denom_idx': None}
```

d

```python
# Extract trained model instance
best_linreg_model.model
```

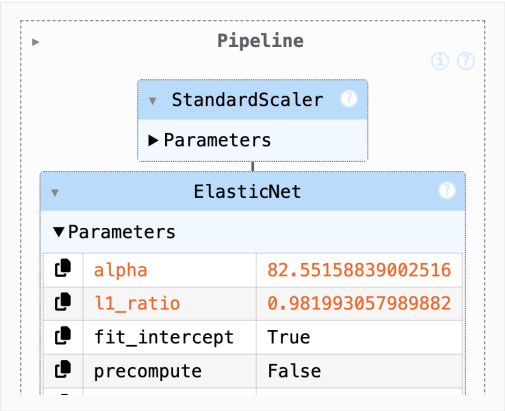

Figure S1: *ritme*'s Python API usage examples for (**a**) finding the best feature-model configuration and the custom-made model object returned, (**b**) prediction on any dataset with the same feature structure, (**c**) extraction of the optimal feature engineering approach and (**d**) the trained estimator.

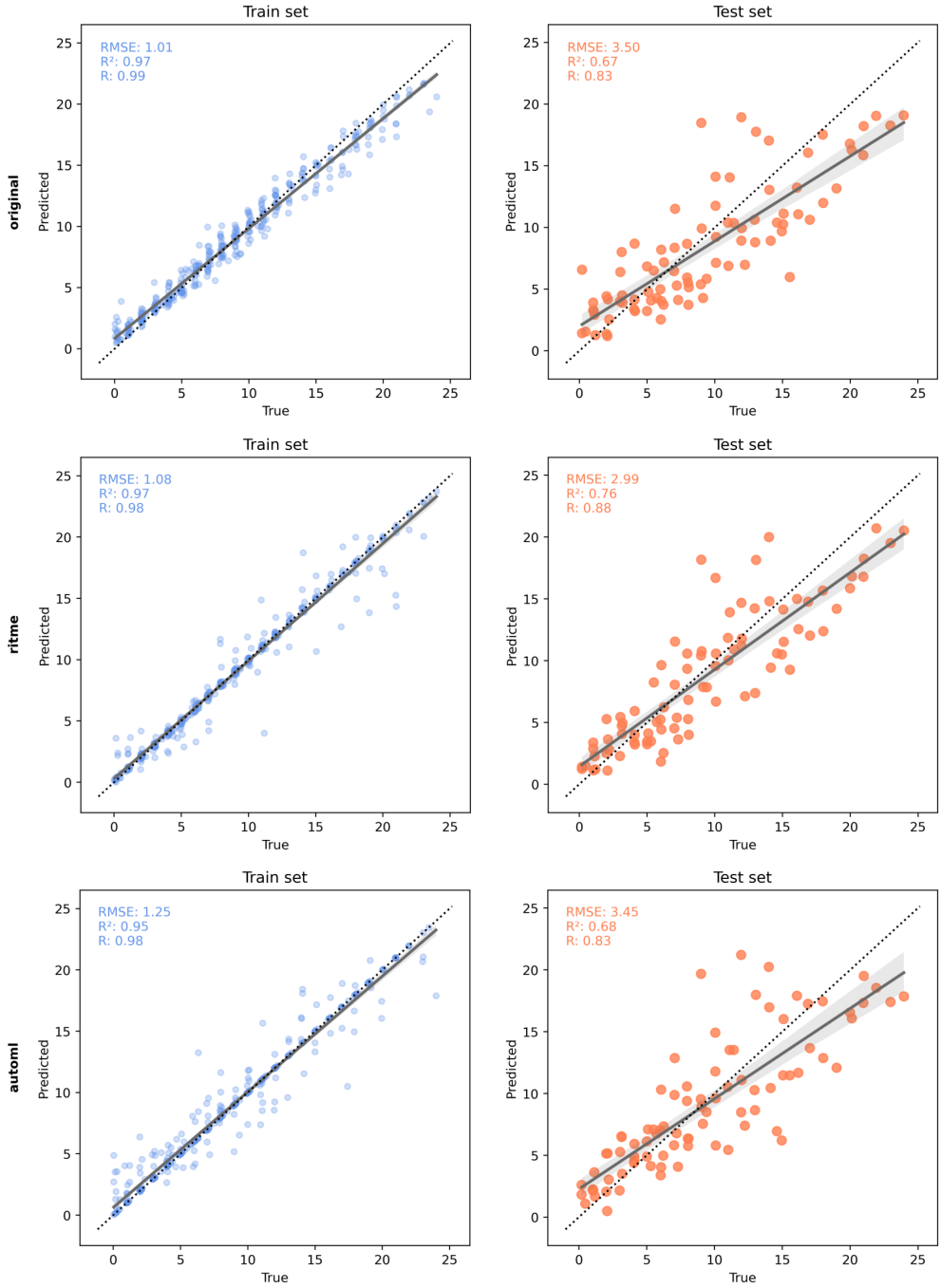

Figure S2: Predictive performance of the study's original model set-ups, *ritme*'s best feature-model configuration and *auto-sklearn*'s matched best model evaluated on the same train-test splits for use case 1 (Table 4). RMSE = root mean squared error, R^2 = coefficient of determination, R = Pearson correlation coefficient.

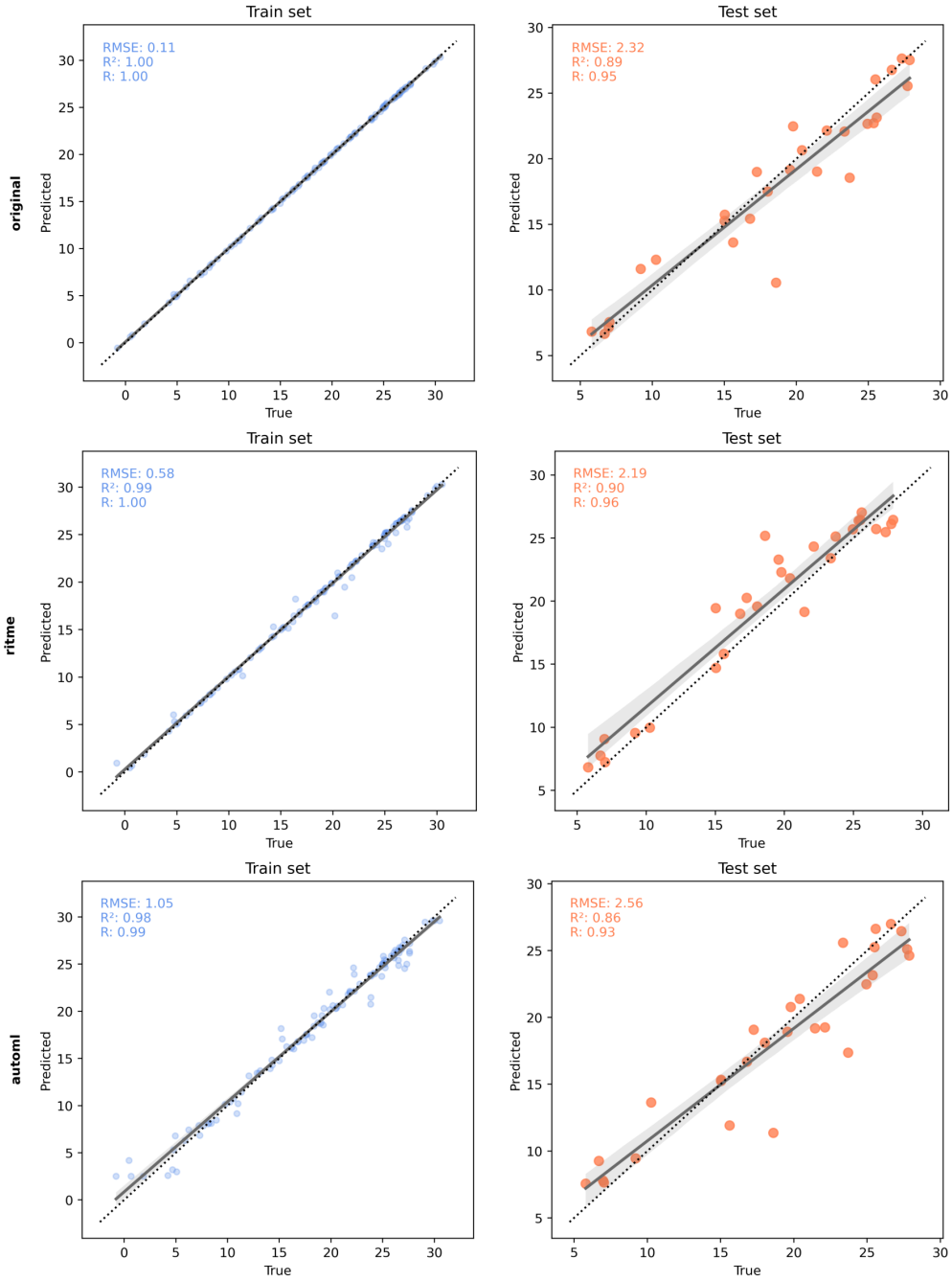

Figure S3: Predictive performance of the study's original model set-ups, *ritme*'s best feature-model configuration and *auto-sklearn*'s matched best model evaluated on the same train-test splits for use case 2 (Table 4). RMSE = root mean squared error, R^2 = coefficient of determination, R = Pearson correlation coefficient.

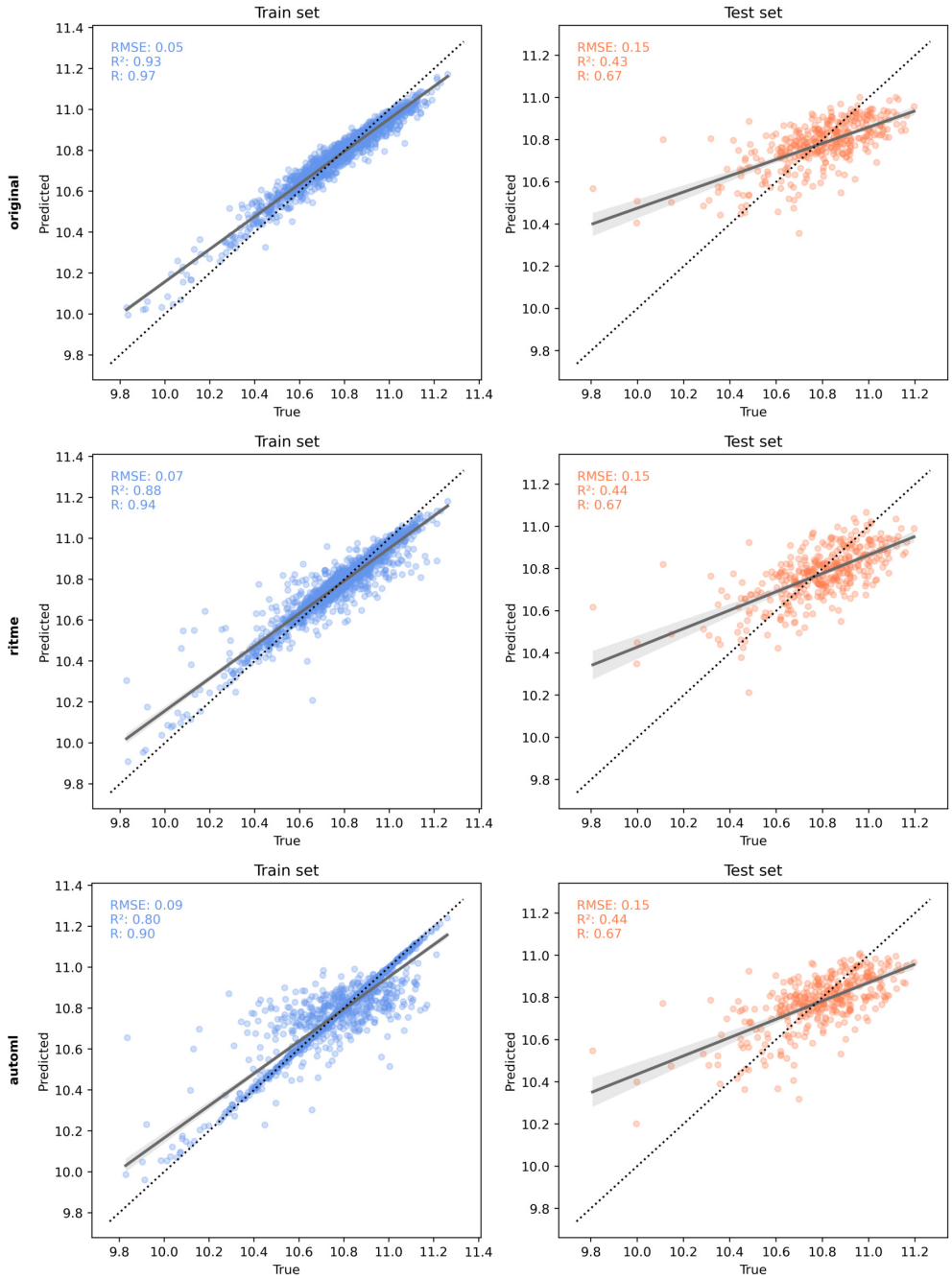

Figure S4: Predictive performance of the study's original model set-ups, *ritme*'s best feature-model configuration and *auto-sklearn*'s matched best model evaluated on the same train-test splits for use case 3 (Table 4). RMSE = root mean squared error, R^2 = coefficient of determination, R = Pearson correlation coefficient.