Dawkins and Gerber *Microbiome (2026)14:21*
https://doi.org/10.1186/s40168-025-02270-z

Microbiome

SOFTWARE

Open Access

# MMETHANE: interpretable AI for predicting host status from microbial composition and metabolomics data

Jennifer J. Dawkins[1,2] and Georg K. Gerber[1,2,3,4*]

## Abstract

**Background** Metabolite production, consumption, and exchange are intimately involved with host health and disease, as well as being key drivers of host-microbiome interactions. Despite the increasing prevalence of datasets that jointly measure microbiome composition and metabolites, computational tools for linking these data to the status of the host remain limited.

**Results** To address these limitations, we developed MMETHANE, a purpose-built deep learning model for predicting host status from paired microbial sequencing and metabolomic data. MMETHANE incorporates prior biological knowledge, including phylogenetic and chemical relationships, and is intrinsically interpretable, outputting an English-language set of rules that explains its decisions. Using a compendium of six datasets with paired microbial composition and metabolomics measurements, we showed that MMETHANE always performed at least on par with existing methods, including blackbox machine learning techniques, and outperformed other methods on 80% of the datasets evaluated. We additionally demonstrated through two cases studies analyzing inflammatory bowel disease gut microbiome datasets that MMETHANE uncovers biologically meaningful links between microbes, metabolites, and disease status.

**Conclusions** MMETHANE is an open-source software package that brings state-of-the-art interpretable AI technologies to the microbiome field, emphasizing usability with simple written explanations of its decisions and biologically relevant visualizations. This robust and accurate tool enables investigation of the interplay between microbes, metabolites, and the host, which is critical for understanding the mechanisms of host-microbial interactions and ultimately improving the diagnosis and treatment of human diseases impacted by the microbiome.

## Introduction

The human gut microbiota is an extremely complex ecosystem that performs a range of critical functions for the host [1–3]. Alterations affecting functioning of the microbiome have been associated with many human diseases and disorders including, allergies, inflammatory bowel disease (IBD), cardiovascular disease, chronic kidney disease, metabolic diseases such as diabetes, and infections [3–5]. Although many such associations have been reported, understanding of the specific molecular mechanisms through which the microbiome influences host health or disease remains limited.

*Correspondence:
Georg K. Gerber
ggerber@bwh.harvard.edu
[1] Division of Computational Pathology, Brigham and Women's Hospital, Boston, MA, USA
[2] Harvard-MIT Health Sciences and Technology, Cambridge, MA, USA
[3] Harvard Medical School, Boston, MA, USA
[4] Massachusetts Host-Microbiome Center, Boston, MA, USA

Metabolites represent one very important mechanism for microbe-microbe and host-microbiome crosstalk. In the gastrointestinal tract in particular, metabolic activity is intense, with production, absorption, and modification of a myriad of compounds by both microbial and host cells [1, 6]. Indeed, strong mechanistic links have been established between some human diseases and microbially derived metabolites, such as promotion of cardiovascular disease by trimethylamine N-oxide [7]; protection against intestinal inflammation in IBD and *Clostridioides difficile* infection (CDI), by butyrate [8]; and protection against CDI by amino acid depletion [9, 10]. These well-studied metabolites likely represent the tip-of-the-iceberg, with high-throughput metabolomics and metagenomics methods offering promise as platforms for discovery of novel metabolite-mediated relationships between the microbiome and host health and disease. Such data, which is high dimensional, noisy, and often has low samples sizes, requires rigorous computational analyses [11]. Moreover, to be most useful for discovering and ultimately understanding underlying mechanisms of host-microbial interactions, it is import that computational analysis methods produce human-interpretable results [12].

Existing computational methods for discovering relationships between high-throughput metabolomics/metagenomics data and host status have several shortcomings. One of the most popular methods is univariate hypothesis testing [4, 13–16], which evaluates differences between cases and controls for each microbial or metabolic feature of interest. These analyses, although straightforward to implement and interpret, consider each feature individually, and thus cannot identify relationships that arise from interactions among multiple features [17], and also have limited statistical power due to the need for multiple hypothesis correction [18]. Supervised machine learning methods are an alternative, which unlike univariate hypothesis testing approaches, can consider all features simultaneously and predict unseen data [17]. Examples of supervised machine learning methods that have been used to analyze metabolomics and metagenomic data include lasso logistic regression (LR) and random forests (RFs) [5, 19, 20]. Popular supervised machine learning methods that have been used extensively in other fields include boosting and deep neural networks (DNNs). An important trade-off for many supervised machine learning methods is their ability to capture complex, nonlinear relationships that may occur in biological systems versus the degree to which their decisions are human-interpretable. For example, the nonlinear methods RFs, boosting, and DNNs, are not directly interpretable, requiring post hoc analyses to understand model choices. These analyses are not faithful to the underlying models and offer explanations that may be oversimplifications or distortions of how the model maps input features to predictions [21].

To address the above-mentioned challenges for linking metabolomic and metagenomic measurements to host status, we developed a computational method, MMETHANE (Microbes and METabolites to Host Analysis Engine), which we make available as an open-source software package. MMETHANE builds on our prior supervised machine learning methods, MITRE and MDITRE [22, 23], which learn human-interpretable rules for predicting a binary label (host status) from longitudinal microbial sequencing data (16S rRNA amplicon or shotgun metagenomics) and the phylogenetic relationships among the taxa. The key advance over our previous methods is that MMETHANE also analyzes metabolomic data and does so *jointly* with microbial composition data. To accomplish this, we modeled inter-metabolite relationships using structure-based chemical distance measures. This domain-specific knowledge enables the model to perform adaptive, biologically relevant dimensionality reduction, yielding interpretable predictors that can also potentially increase statistical power in the setting of noisy data and low sample sizes.

The remainder of this manuscript is organized as follows. We first present the MMETHANE method, describing the purpose-built model and its incorporation of domain-specific knowledge. Next, we detail the compendium of six datasets we compiled to benchmark MMETHANE and comparator methods. Using these datasets, we assess five structure-based chemical similarities for MMETHANE, to determine which is most informative for use in subsequent benchmarking. We then compare predictive performance of our method against four other supervised learning methods—LR, RFs, AdaBoost, and DNNs—on the dataset compendium. To further understand observed differences in predictive performance between the methods, we benchmark them on semi-synthetic data with varying sample sizes and types of associations that may occur in real data. Finally, we assess MMETHANE's ability to uncover biologically meaningful relationships among metabolites, microbes, and the host, through two case studies.

## Results

### MMETHANE is a purpose-built deep learning method provided as an open-source software package that infers human-interpretable rules predicting host status from microbial composition and metabolomics data

MMETHANE is a supervised machine learning method that predicts host status (e.g., disease or no disease) from paired microbial composition and metabolomic data (Fig. 1A). The MMETHANE model captures non-linear

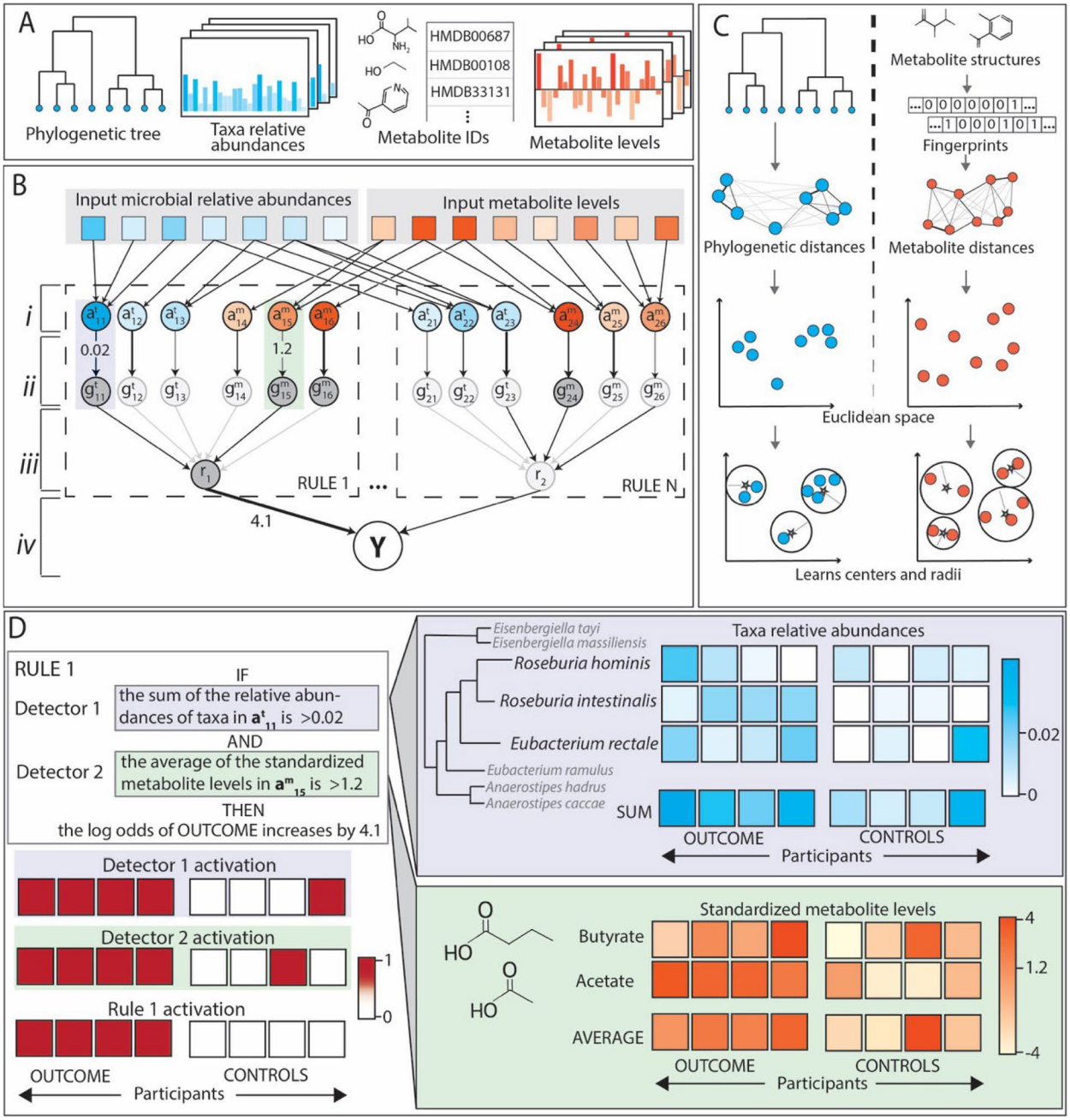


**Fig. 1** MMETHANE is a purpose-built deep learning method that infers human-interpretable rules predicting host status from microbial composition and metabolomics data. **A** Inputs to MMETHANE are a phylogenetic tree, microbial relative abundances, metabolite IDs (from which chemical structures are automatically retrieved), and metabolite levels. **B** Schematic of the custom deep learning architecture. Input taxa and metabolites are processed by a feedforward deep neural network with four layers: *(i)* feature focus, aggregates input features (taxa or metabolites) into detectors, *(ii)* detector activation, "on" (dark grey) if the input from the previous layer surpasses a threshold, *(iii)* logical-AND layer, combines detector activations into a final rule activation, and *(iv)* prediction layer, outputs the probability of the host status label based on the activations and strengths of the input rules. The model probabilistically selects which detectors and rules to include, with bias toward sparsity (depicted here by the darkness of edges). **C** Schematic of how feature focus layers use phylogenetic or metabolite distances to learn groupings for detectors. Distances, derived either from sequences for taxa or chemical structures for metabolites, are embedded into Euclidean spaces, and each detector learns a center and radius that defines the grouping it operates on. **D** Example of how the learned model structure is directly interpretable as English-language rules. Edges from the inputs → layer (*i*) define which taxa or metabolites participate in each detector; edges from layer (*i*) → (*ii*) define the detector threshold; edges from layer (*ii*) → (*iii*) define which detectors are used; edges from layer (*iii*) → (*iv*) define which rules are used, and the edge weights define the contribution of the rule to the log odds of the host status label

interactions while maintaining human-interpretability through a custom architecture that learns sparse sets of English-language *rules*. As shown in Fig. 1B, MMETHANE uses a feedforward deep neural network with four layers. The parameters learned by the layers can be directly interpreted as components of rules. Each rule consists of a set of *detectors*. Each detector learns: (a) a group of microbial taxa or metabolites and aggregation operation for the group, i.e., sum of relative abundances of taxa or averages of standardized metabolite levels (layer *i*), and (b) an activation threshold, i.e., the detector is *on* if the aggregated value for the group surpasses the threshold (layer *ii*). Activations for the detectors are combined via a logical-AND to form the activation for each rule (layer *iii*). Finally, a prediction for the status of each host is calculated from a weighted combination of rule activations (layer *iv*). To summarize, each rule combines multiple detectors via a logical-AND structure and the entire predictor combines multiple rules via a weighted logical-OR structure. MMETHANE uses a Bayesian framework to bias the model toward sparsity, i.e., learning small numbers of detectors and rules. Because efficiently learning parameters in rule-based models is generally very computationally intensive [23], we developed an efficient inference method based on mathematical relaxations of discrete-valued functions. See "Methods" section for details.

As with our previous work on rule-based models for microbiome data analysis [22, 23], we designed the MMETHANE model to leverage prior knowledge to learn biologically informed groupings (Fig. 1C) of input features (taxa or metabolites). The matrix of pairwise distances between input features are embedded in a Euclidean space, and then each detector learns a center and radius in this space. Inclusion in the group is then determined using a relaxed set inclusion distance-based function that we developed [23]. Distances for taxa are based on phylogenetic trees, as in our previous work [22, 23]. For metabolites, we developed a framework for modeling distances based on molecular fingerprinting approaches. We investigated a range of fingerprinting approaches and include the capability in the MMETHANE software to calculate five different fingerprints from input metabolite IDs. See "Methods" section and the subsequent section on empirical choice of an optimal fingerprinting approach for additional details.

We provide MMETHANE to the community as an open-source Python package. To run an analysis using the software, the user supplies four files as inputs: (1) a phylogenetic tree (or a default tree may be used), (2) a table of microbial counts or relative abundances, (3) a table of metabolite IDs, and (4) a table of continuous-valued metabolite levels. See "Methods" section and the software documentation for a complete description of the input formats. The outputs of the MMETHANE software are: (1) a binary prediction for each input subject, (2) a set of English-language rules explaining model decisions, and (3) visualizations for each rule. Examples of rules and their visualizations are shown in Fig. 1D and in the subsequent case-study sections.

### Compendium of paired microbial composition and metabolomics data

To facilitate benchmarking of MMETHANE and state-of-the-art methods, we compiled a compendium of publicly available datasets containing paired microbial sequencing (either 16S rRNA amplicon or shotgun metagenomics) and metabolomics measurements. Most datasets included in the compendium were part of a curated gut microbiome-metabolome data resource [24]. To ensure bioinformatics consistency, microbial sequencing data was reprocessed using dada2 for 16s rRNA amplicon data [25], and Metaphlan3 for metagenomics data [26]. We limited datasets chosen to those with at least two distinct outcome/treatment groups, as our goal was to assess predictive accuracy of methods. This yielded eight potential datasets. After running benchmarking (see below), we discarded for further consideration any datasets in which no method achieved an area under the receiver-operator curve (AUC) > 0.6 on either metabolomics or microbial sequencing data (or combined), as those datasets were deemed to have too little signal to be used to compare performance meaningfully across methods.

This selection procedure yielded six datasets (out of the original eight considered) for the compendium: He et al. [13], Dawkins et al. [5], Erawijantari et al. [16], Lloyd-Price et al. [14], Franzosa et al. [15], and, Wang et al. [4]. Briefly, He et al. analyzed fecal samples from breastfed and formula-fed infants using H1 NMR metabolomics and 16S rRNA amplicon sequencing [13]. Dawkins et al. analyzed fecal samples of patients diagnosed with *Clostridioides difficile* infection (CDI) after completion of antibiotic treatment using 16S rRNA amplicon sequencing and LC–MS metabolomics [5]. Erawijantari et al. analyzed fecal samples from 42 participants who had previously undergone gastrectomy and 54 healthy controls using shotgun metagenomics and time-of-flight MS (TOF–MS) [16]. Lloyd-Price et al. analyzed fecal samples with LC–MS metabolomics and shotgun metagenomics from 79 IBD patients and 26 healthy controls [14]. Franzosa et al. used the same methodology as Lloyd-Price et al. to analyze fecal samples from 121 IBD patients and 34 healthy controls [15]. Wang et al. performed HPLC–MS/MS metabolomics and shotgun metagenomic sequencing on fecal samples from 220 patients with end-stage renal disease (ESRD) and 67 healthy controls [4].

Additional details on each dataset can be found in Supplementary Table S1.

### Benchmarking on the data compendium selected the PubChem CACTVS fingerprint as the best metabolite measure for MMETHANE

To assess the influence of the molecular fingerprinting measures on predictive performance in MMETHANE, we evaluated five fingerprints: PubChem CACTVS [27], Morgan, MAP4 [28], MQN [29] and InfoMax [30]. We chose these fingerprints based on their prevalence of usage in the literature and to span a range of complexities and approaches, including molecular fingerprints based on substructure features, metabolite topology, and deep learning representation of metabolite features. See "Methods" section and Supplementary Table S2 for additional details on the fingerprints we evaluated. Similarities were calculated from fingerprints using the Tanimoto/Jaccard metric for the binary PubChem CACTVS and Morgan fingerprints, and the city block distance for the MQN fingerprint; for the MAP4 and InfoMax fingerprints, we used the similarities provided by their respective software packages. For each fingerprinting method, we evaluated MMETHANE's fivefold cross-validated predictive performance (AUC) on the data compendium.

Overall, the PubChem CACTVS fingerprint provided the most consistent performance (Supplementary Fig. S1). It was the top performer in 3/6 datasets: Dawkins et al., Erawijantari et al., and Fransoza et al. For the other three datasets, although there were a few instances in which performance with the PubChem CACTVS fingerprint was not significantly different than with other fingerprints, no other fingerprint outperformed it with statistical significance. Thus, for efficiency and compactness of our presentation, we employ the PubChem CACTVS fingerprint for the subsequent analyses described. However, as described above, we provide the option in the MMETHANE software for users to easily calculate similarities with any of the other methods, to assess performance on their own data.

### MMETHANE predicted host status as well or better than state-of-the-art machine learning methods on the data compendium

We benchmarked MMETHANE's predictive performance on all datasets in the compendium against that of four popular supervised machine learning methods: lasso logistic regression (LR), Random Forests (RF), AdaBoost, and a feed-forward neural network (FFNN). LR was chosen as a comparator method because it has been widely used for analyzing microbiome data and is inherently interpretable [5, 19, 20]. RF, a nonlinear method, has also been widely used for analyzing microbiome data, including on datasets with paired microbial sequencing and metabolomics measurements [5, 19, 20]. AdaBoost and FFNN are exemplars of additional types of nonlinear methods, and are very popular in the broader machine learning field, with some applications in the microbiome field, as well [31–37]. We evaluated performance using cross-validated AUC and on three different prediction tasks: metabolites only, taxa only, and metabolites and taxa combined. Five-fold cross-validation was performed on all datasets, except for Dawkins et al. [5], which had the smallest number of participants. For this dataset, we used leave-one-out-cross-validation, which does not provide an estimate of variance across folds; overall, results for this dataset should therefore be interpreted with more caution. Note that the nonlinear methods RF, AdaBoost, and FFNN are not inherently interpretable, and post hoc approaches for their interpretation are not generally faithful to the underlying models, which can result in incorrect inferences [38]. Also of note, MMETHANE and many of the other models evaluated could in principle incorporate co-variates. However, the number of available co-variates, their documentation and relevance to label of interest, and sparseness (i.e., missing data) varied considerably in the datasets in our compendium. Thus, because our goal was to provide as uniform and interpretable benchmarking across the datasets as possible, and to focus on signal provided by microbial abundances and metabolites, we chose not to include co-variates in the benchmarking analyses.

Overall, MMETHANE performed on par or outperformed all the comparator methods (Fig. 2), across all prediction tasks. For predicting from metabolomics data only, MMETHANE was the top performer on 4/6 datasets. On microbial composition data only, MMETHANE and FFNN were top performers on 3/4 datasets (note that 2 datasets were excluded from this analysis, because no method achieved an AUC above 0.6). On combined metabolomic and microbial composition data, MMETHANE was the top performer on 5/6 datasets. Notably, MMETHANE outperformed the only other directly interpretable method, lasso LR, in all cases except for predicting CDI recurrence from metabolomics data only. To assess the utility of pre-defined groupings based on biological information, we also evaluated the comparator methods with inputs of data aggregating family-level taxa or class-level metabolites (Supplementary Table S3). Overall, we found that models given non-aggregated data performed significantly better than those given aggregated data. Indeed, there was no case in which a model trained on aggregated data performed better than a model trained on non-aggregated data. These results suggest that these types of strict, pre-defined groupings are

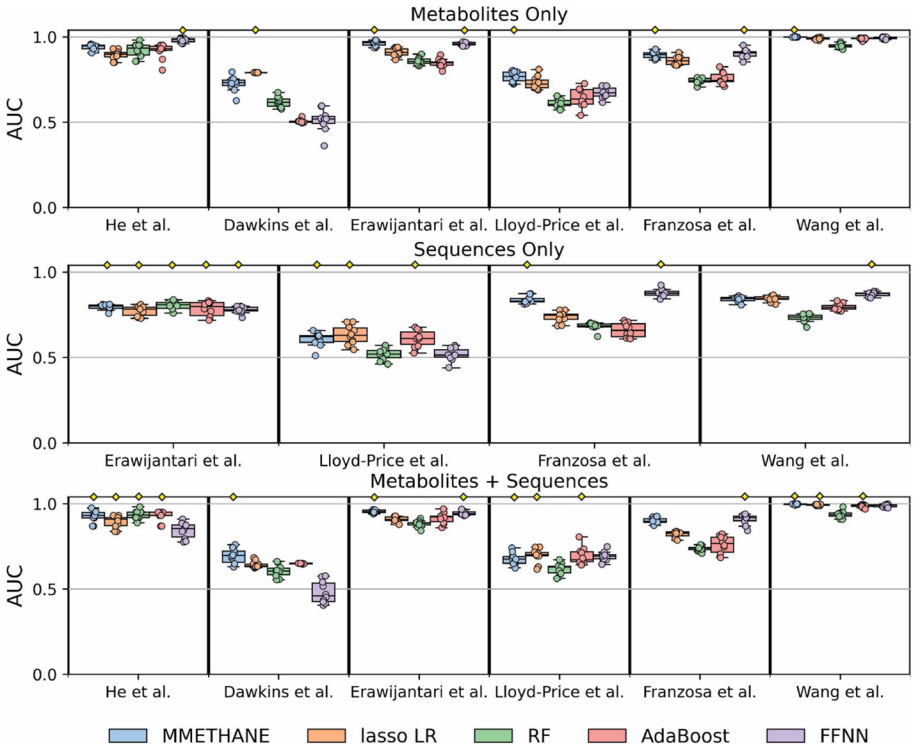


**Fig. 2** MMETHANE predicted host status on par or better than all state-of-the-art comparator methods on the compendium of microbial sequencing and metabolomics data. Five-fold cross-validated AUC scores for prediction of host status on the six datasets in the compendium are shown, for MMETHANE, lasso logistic regression (LR), random forest (RF), adaptive boosting (AdaBoost), and a feed forward neural network (FFNN). Methods were evaluated with inputs of metabolomic data alone, sequence data alone, or both. Box plots with results from ten random seeds, with medians and 95% intervals, are shown. Yellow diamonds indicate the top score or scores (if multiple scores were not significantly different from the top score)

too restrictive and the ability to learn groupings more flexibly is important for optimal predictive performance.

Our benchmarking results also demonstrate that metabolites were more predictive of host status than microbial taxa compositions in these datasets. Moreover, in almost all cases, predictive performance when both inputs were provided was not significantly better than performance when metabolites only were provided. This result is consistent with prior observations regarding the predictiveness of the gut metabolome versus microbial composition [5]. Interestingly, however, on 5/6 of the datasets, at least one method used both metabolite and microbial composition predictors, and on half the datasets, all methods used both types of predictors (Supplementary Table S4). These results demonstrate that metabolite data was overall more predictive of host status on the datasets assessed, although microbial composition data also provided meaningful signal in many cases; however, these results must be interpreted in the context of the machine learning models and datasets that we assessed, and it is possible that future models, or larger or more diverse datasets, could reveal additional signal from microbial composition information alone or in combination with metabolites.

### MMETHANE exhibited consistently superior performance across sample sizes and types of associations compared to state-of-the-art methods as assessed on semi-synthetic data

We next sought to understand the basis for MMETHANE's strong performance on the data compendium. Factors that we reasoned could be responsible for this included MMETHANE's ability to detect combinations of biologically plausible associations, and data efficiency (ability to perform well with small sample sizes while continuing to increase performance with more data). An analysis of these factors requires ground truth knowledge of associations, so we employed a semi-synthetic data generation approach that artificially introduced differences into data from the control group of a reference

dataset. For the reference dataset, we used Wang et al. [4], because it had the largest number of control samples in the data compendium. The dataset generation procedure that we used was designed not only to capture realistic aspects of the data compendium we analyzed, but also to test performance on aspects of data that could be encountered in future studies, such as larger numbers of subjects or synergistic signal between microbes and metabolites.

Briefly, our semi-synthetic dataset generation and evaluation procedure was as follows. For each simulation, half the participants were selected (uniformly at random, with replacement) to be "cases" that received the perturbation and half were selected (uniformly at random, with replacement) as "controls" that did not. Next, a group (or groups) of metabolites and/or microbial taxa were selected uniformly at random to be perturbed using parameters derived from the original data (e.g., effect sizes of perturbations). We considered five different perturbation scenarios that assessed methods' abilities to predict from metabolomic or taxa data separately, or in combination (Fig. 3). The first and third scenarios, which involved perturbing either a single metabolite group or a single taxonomic clade, were expected to be the easiest cases for the algorithms to learn. The remaining scenarios included combinations of associations (e.g., as shown in Fig. 3A, for associations with both microbial compositions and metabolites), emulating biological phenomena that involve nonlinear interactions such as allostery [39]. These types of relationships were expected to be more difficult for the algorithms to learn; in particular, logistic regression, which does not model interactions among variables, was expected to perform poorly on these cases. For all the scenarios, we assessed predictive performance on datasets of varying sample sizes. We simulated data with 36, 48, 64, 128, 300, and 1000 participants; this range was selected based on realistic dataset sizes with the lower and upper values smaller or larger than those of the datasets analyzed. We evaluated MMETHANE against the four comparator methods and using the same cross-validation strategy as for the real data as described above. See "Methods" section for full details on the semi-synthetic data generation procedure. Of note, simulated data with 128, 300, and 1000 participants required bootstrapping baseline microbiomes, because the original dataset contained only 67 controls, and should thus be interpreted with more caution because performance estimates could potentially be over-optimistic (although not favoring any particular comparator method).

MMETHANE outperformed or was on par with all the other methods on all the semi-synthetic data scenarios (Fig. 3B). Some general trends emerged, which were useful in understanding performance differences among the methods. First, comparing across scenarios with metabolomic data, taxa composition data, or both, we saw that all methods generally performed better on the metabolite data alone. This was consistent with our results on real data and indicated that our semi-synthetic data generation procedure captured this phenomenon. Second, even on the simplest scenarios, of perturbation of a single group of metabolites or a single taxonomic clade, MMETHANE remained the sole top performer up to a sample size of 128 subjects, after which it shared the top spot with multiple methods. These results indicate that MMETHANE was more data efficient than the other methods, likely because it not only encodes penalties on model complexity, but also incorporates biological knowledge through known phylogenetic or chemical relationships. Third and finally, as expected, all the methods, including MMETHANE, made less accurate predictions on the more complex scenarios that involved interactions between sets of taxa, metabolites, or both. Also, as expected, LR performed poorly on these scenarios, even with large sample sizes, because it is incapable of capturing nonlinear interactions. Interestingly, methods capable of capturing nonlinear interactions other than MMETHANE did not show consistent improvements in performance with larger sample sizes, as they had on the simpler

(See figure on next page.)

**Fig. 3** MMETHANE exhibited consistently superior performance across sample sizes and types of associations compared to state-of-the-art methods assessed on semi-synthetic data. A semi-synthetic data generation approach was used that introduced simulated differences into a real reference dataset. Five scenarios were created to assess methods' abilities to predict from metabolomic or taxa data separately, or in combination, and in each scenario the sample size was varied to assess data efficiency of the methods. **A** Schematic showing the simulation process for scenario 5, the most complex, in which 1 metabolite and 1 taxa group were perturbed. Participants from the control group of a reference real dataset were randomly assigned to synthetic "control" and "case" groups. Members of the "control" group then received either the metabolite group or the taxa group perturbation, but not both, whereas members of the "case" group received both types of perturbations. Scenarios 2 and 4 were generated similarly, except they involved 2 groups of either metabolites or taxa, but not both. Scenarios 1 and 3 involved a single group of either metabolites or taxa. **B** Five-fold cross-validated AUC scores for prediction of host status for the five scenarios are shown, for MMETHANE, lasso logistic regression (LR), random forest (RF), adaptive boosting (AdaBoost), and a feed forward neural network (FFNN). Ten random semi-synthetic datasets were generated for each scenario and sample size, and models were run with ten random seeds. Box plots indicate medians and 95% intervals. Yellow diamonds indicate the top score or scores (if multiple scores were not significantly different from the top score)

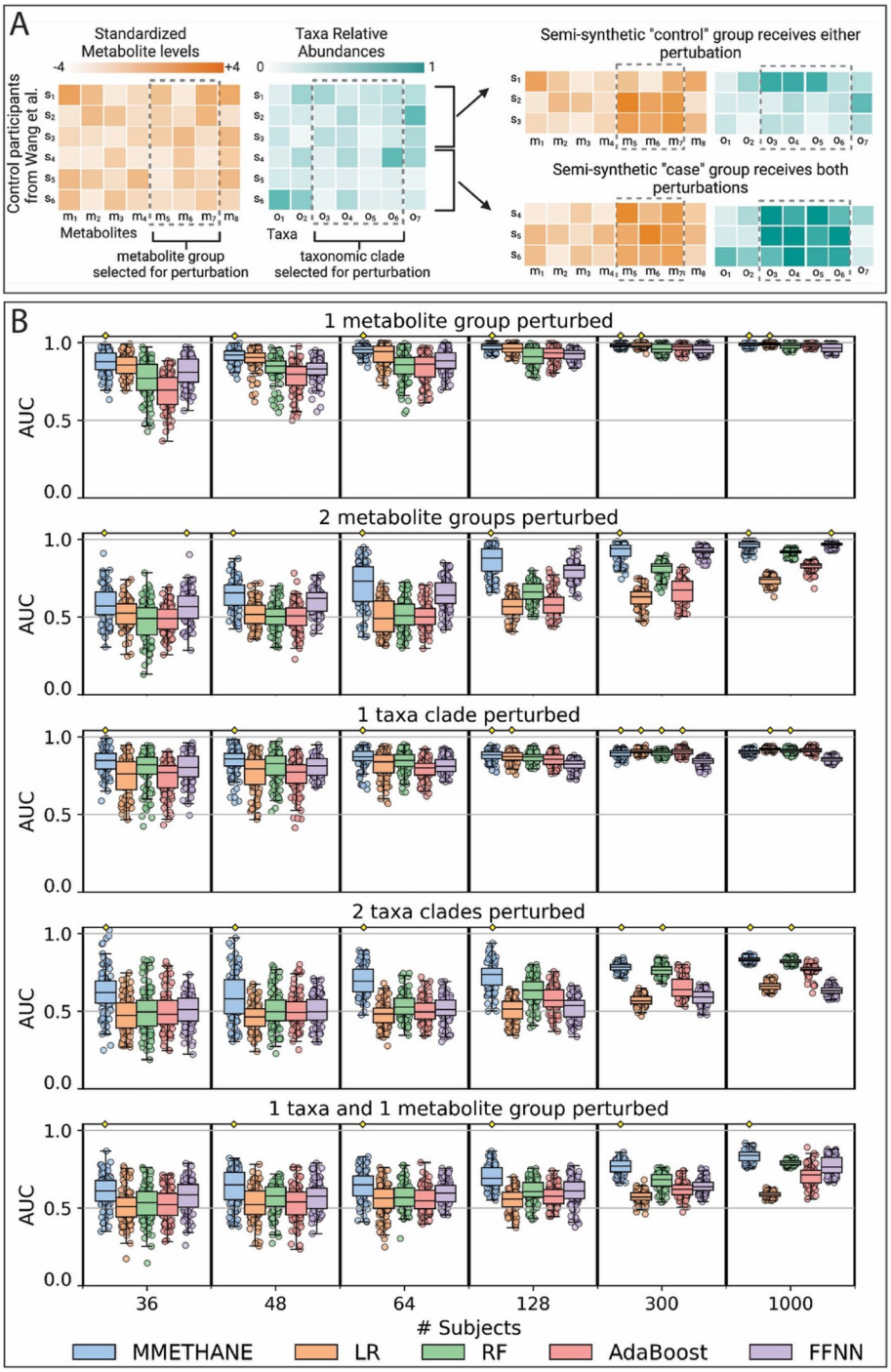


**Fig. 3** (See legend on previous page.)

scenarios. The goal of our simulation procedure was to generate increasingly difficult scenarios, to gain insights into what scenarios led to performance degradation. On these benchmarks, MMETHANE inferred non-linear interactions better than comparator methods and its performance degraded more slowly than comparator methods. However, as expected, we saw degradation of performance as the complexity of the scenario increased, providing insights into the limits of what MMETHANE can infer from sufficiently noisy and sparse datasets.

### Case studies: MMETHANE learned rules linking gut bacterial taxa and metabolites to inflammatory bowel disease in the host

After demonstrating that MMETHANE performed on par or outperformed other supervised machine learning methods on real and semi-synthetic data, we next investigated our method's ability to discover biologically meaningful associations. We used data from two analyses of gut microbes and metabolites from participants with IBD as our case studies. Our rationale for choosing these studies was the clinical importance of IBD and its complexity as a pathological entity, with the rationale that MMETHANE's ability to group taxa or metabolites and find interpretable non-linear relationships between them could provide additional insights into underlying mechanisms of microbe, metabolite, and host interactions in the disease.

#### *Case Study 1: Lloyd-Price et al*

MMETHANE found two rules on this dataset (Fig. 4A, B), which analyzed fecal samples with LC–MS metabolomics and shotgun metagenomics from 79 participants with IBD and 26 healthy controls. The first rule, which contains a single detector, states that if the aggregated relative abundance of a group of 17 taxa is increased above the learned threshold, the odds of the sample being classified as IBD is decreased by $3 \times 10^5$ fold (Fig. 4A). Many of the taxa have previously been associated with anti-inflammatory activity and intestinal barrier integrity [7, 40], and are producers of short chain fatty acids, which are known to mediate these effects [8, 41, 42]. This rule thus identifies taxa with a biologically plausible protective role in IBD, consistent with its semantics that higher levels of these taxa are associated with lower odds of IBD.

The second rule contained one metabolite detector and one taxa detector, and states that if the aggregated values of both detectors are above the learned thresholds, the odds of the sample being classified as IBD is increased by 24-fold (Fig. 4B). The metabolite detector in this rule groups together three compounds, tryptophan, 5-hydroxy-tryptophan, and serotonin, which, are the substrate, intermediate, and product of serotonin biosynthesis from tryptophan, respectively. Human and animal studies have shown that increased levels of serotonin resulted in increased severity of intestinal colitis through inhibition of autophagy and beta-defensin production in gut epithelial cells [43, 44], providing a possible mechanism explaining this detector's utility in predicting IBD. The microbial detector in this rule identified a group of four taxa, *Anaerotruncus colihominis, Faecalibacterium prausnitzii, Gemmiger formicilis*, and *Ruthenibacterium lactatiformans*. Although these taxa are SCFA producers [41, 45, 46], which as discussed for rule one, typically are associated with protection against IBD, rule two suggests these taxa may also play a more nuanced role in IBD in the context of increased serotonin levels in the gut. Interestingly, microbially-derived SCFAs have been found to upregulate *Tph1* and thus increase serotonin synthesis in gut epithelial cells [44]. Additionally, tryptophan (the substrate of serotonin biosynthesis) is metabolized by bacteria in the gut, including, *Ruminococcus gnavus* and *Eubacterium halii* [47, 48], which were identified in the first rule. These findings thus suggest a possible link between serotonin biosynthesis in the gut and microbial activities that may modulate both substrate availability as well as serotonin production in host epithelial cells and its promotion of inflammation.

#### *Case Study 2: Franzosa et al.*

MMETHANE found a single rule for this dataset (Fig. 4C), which analyzed fecal samples with LC–MS metabolomics and shotgun metagenomics from 121 participants with IBD and 34 healthy controls. The rule,

(See figure on next page.)

**Fig. 4** MMETHANE learned interpretable rules with biologically relevant taxa and metabolites for classifying inflammatory bowel disease (IBD) versus healthy controls. Visualizations automatically produced by the software are shown. Each visualization includes the phylogenetic relationships of the identified taxa and the chemical structures of the identified metabolites, as well as a display of the underlying data and how the rule and detector activations influence the prediction for each participant. **A** and **B** show the two rules MMETHANE learned when applied to the Lloyd-Price et al. dataset, which analyzed fecal samples with LC–MS metabolomics and shotgun metagenomics from 79 participants with IBD and 26 healthy controls. The first rule (**A**) consists of a single detector (clause), which states that if the aggregated relative abundance of a set of 17 taxa, many of which are short-chain fatty acid producers, is greater than the threshold, then the odds of the participant having IBD is lower. The second rule (**B**) consists of two detectors (clauses), which state that if the aggregated relative abundance of a group of 4 taxa is greater than a threshold *and* the average levels of a group of three metabolites is greater than a second threshold, then the odds of the participant having IBD is increased. As discussed in the text, this rule suggests a possible context dependent role for the identified group of bacteria, which may modulate production of pro-inflammatory serotonin. **C** Shows the rule MMETHANE learned when applied to the Franzosa et al. dataset, which analyzed fecal samples with LC–MS metabolomics and shotgun metagenomics from 121 participants with IBD and 34 healthy controls. The rule consists of three detectors (or clauses), which if all are true, predicts the odds of the participant having IBD is decreased. The first two detectors identify single metabolites, and the third detector identifies a group of 35 taxa. As discussed in the manuscript, the rule identifies metabolites and taxa with possible synergistic roles in promoting a non-inflammatory intestinal environment

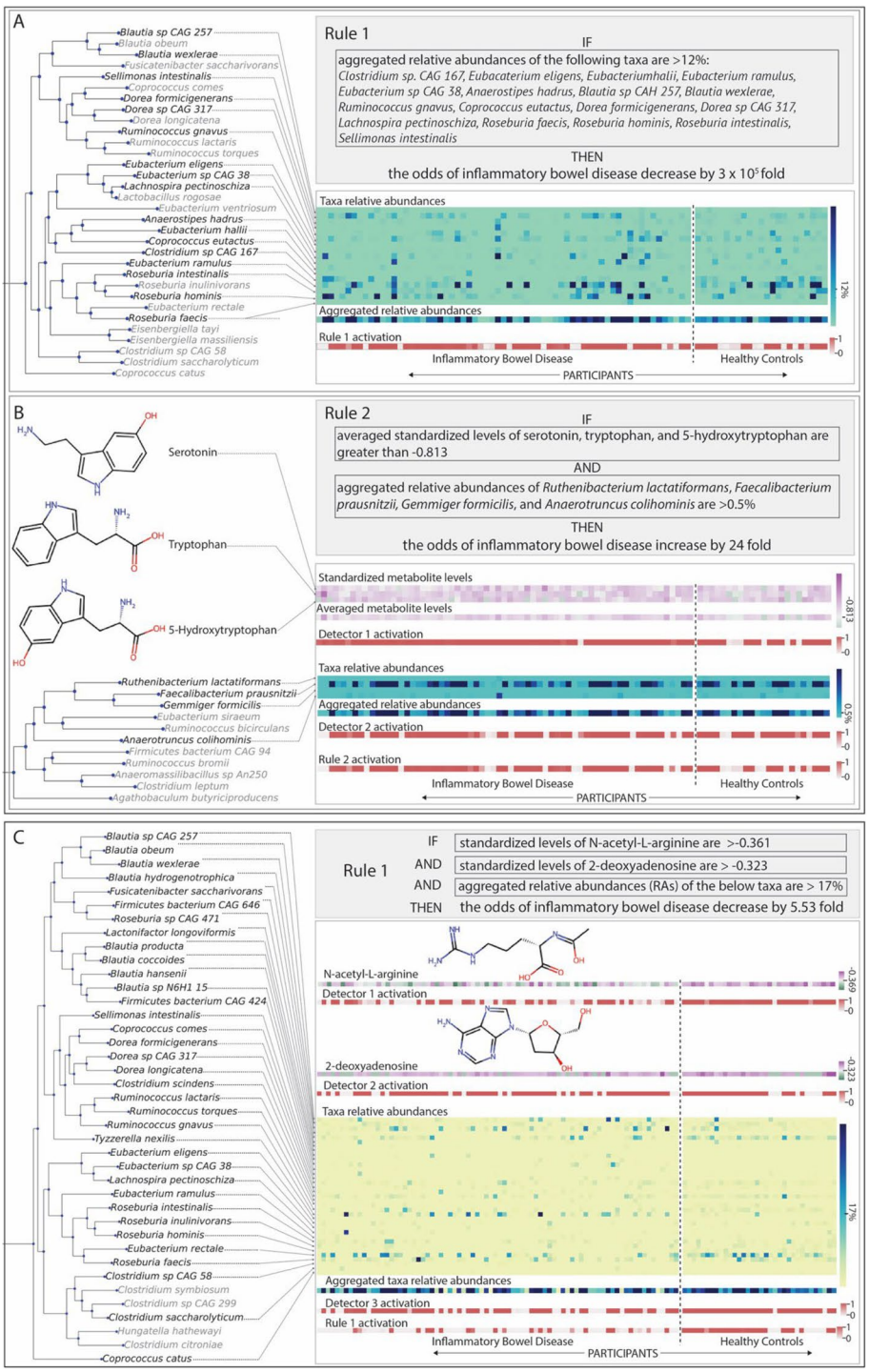


**Fig. 4** (See legend on previous page.)

which contained three detectors (two metabolite and one microbial detector), states that if the aggregated values of all detectors are greater than the learned thresholds, then the odds of the sample being classified as IBD is decreased by 5.5-fold. The first metabolite detector in the rule identified 2-deoxyadenosine, a precursor of dATP (and ultimately DNA), and thus may be a marker of epithelial cell proliferation or commensal bacterial growth. The second metabolite detector in the rule identified N-acetyl-L-arginine, a precursor in arginine

biosynthesis, which may similarly be a marker of healthy cellular growth. Additionally, arginine has been shown to increase intestinal barrier integrity and reduce susceptibility to colitis [49–52]. The microbial detector in the rule identified 35 taxa, including species within the *Blautia, Roseburia, Dorea, Ruminococcus,* and *Eubacterium* genera. A number of these taxa overlap with those in the first rule in case study 1, and are SCFA producers, including *Eubacterium eligens, Roseburia intestinalis, Roseburia hominis, Dorea formicigenerans,* and *Sellimonas intestinalis*. Interestingly, there is a possible mechanistic link between this microbial detector and the second metabolite detector: relative abundances of *Ruminococcus* and *Clostridia* species [50] have been shown to be higher with increased levels of arginine, and two taxa in the microbial detector, *Ruminococcus gnavus* and *Clostridium scindens* are known fermenters of alpha-amino acids, including arginine [10, 48, 53]. Taken together, this rule identified metabolites and taxa with probable synergistic roles in promoting a healthy, non-inflammatory intestinal environment.

## Discussion

We have presented MMETHANE, a purpose-built deep learning method that incorporates domain-specific knowledge to learn human-interpretable rules that predict the status of the host from paired metabolomics and microbial composition data. Our results on a compendium of real data demonstrated that MMETHANE's predictive performance was on par with or surpassed that of state-of-the-art supervised learning methods, including black-box approaches. Analyses on semi-synthetic data moreover demonstrated that MMETHANE was more data efficient than the comparator methods, including on complex scenarios modeling nonlinear relationships among metabolites and/or taxa. Further, we demonstrated that the human-readable rules output by MMETHANE, which automatically group together phylogenetically similar taxa or biochemically-related metabolites, provided biologically relevant insights and suggested interesting hypotheses about the complex interplay between microbes, metabolites, and the host. For example, MMETHANE learned a rule from a dataset with gut metagenomics and metabolomics data from participants with IBD, which suggested a hypothesis that serotonin may have a role in IBD that is modulated by different groups of microbes that either alter substrate availability or have effects on serotonin production in host epithelial cells.

This study has several limitations that suggest future work. First, MMETHANE does not currently incorporate time-series analysis capabilities, as did our previous methods [22, 23] upon which MMETHANE is based. MMETHANE could be readily extended to handle longitudinal datasets; the reason we did not do so in the present study was due to the lack of available paired metabolomics and microbial composition time-series datasets with host outcomes, which would be needed to evaluate our method. Indeed, such datasets would be very valuable for the field overall, as they could provide potentially causal insights into microbe-microbe and host-microbiome interactions, which are difficult to resolve in cross-sectional studies. Second, although not directly related to the MMETHANE model itself, we found that for at least two of the datasets we used as benchmarks, He et al. and Wang et al., the levels of a small number of metabolites (i.e., from specific dietary components, medications, or the disease process itself) almost completely differentiated the participant groups under study. This highlights the need for future studies with larger cohort sizes and designs that allow for explicit control or modeling of confounding factors, which will be important for establishing the relevance of specific gut metabolites to host diseases. Third, MMETHANE is a fully supervised machine learning method, which means that by design it is biased toward finding features, and interactions among those features, that influence host status. For different analysis goals, such as finding features not linked to host status, other approaches, such as unsupervised learning methods, may be more suitable. In future work, it would be interesting to investigate whether adding a generative modeling component to MMETHANE upstream of its prediction layers (e.g., including latent variables that jointly generate the taxa composition and metabolomic data) could effectively balance predictive accuracy with modeling other types of relationships in the data. Fourth and finally, the detector structure of MMETHANE that incorporates biological knowledge, although demonstrably effective on the datasets analyzed, could be extended to capture richer information. For example, MMETHANE currently uses phylogenetic trees constructed from 16S rRNA gene sequences. Future models could incorporate the full gene content of organisms, which could help the method to group together phylogenetically distant but functionally related organisms. Additionally, such information could be layered with knowledge of metabolic pathways, which could allow the model to explicitly link microbes and their metabolic inputs and outputs.

## Conclusion

MMETHANE, which we make available as an open-source software package, is a computational tool for discovering predictive relationships between paired high-throughput microbial sequencing and metabolomics data, and the status of the host. Our method employs a custom-designed deep learning architecture

incorporating biological knowledge to provide accurate predictions without sacrificing interpretability. We thus believe MMETHANE will be a valuable resource for researchers seeking to elucidate the interplay between microbes, metabolites, and the host.

## Methods

### Resource availability

#### *Lead contact*

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Georg K. Gerber (ggerber@bwh.harvard.edu).

#### *Materials availability*

This study did not generate any unique materials or reagents.

#### *Data and code availability*

The MMETHANE model is available at https://github.com/gerberlab/mmethane [54] and has been deposited in Zenodo (https://doi.org/10.5281/zenodo.14058421). Code to reproduce all analyses and figures in this project, as well as the semi-synthetic and real datasets used to produce results is available at https://github.com/gerberlab/mmethane-paper-results [55] and has been deposited in Zenodo (https://doi.org/10.5281/zenodo.14058421). The MMETHANE software is available for use under the GNU General Public License v3. Data needed to reproduce all figures is available at https://doi.org/10.5281/zenodo.14064092. See Table 1 for a summary of availability for all code and data used in the manuscript.

## Method details

### MMETHANE software package

MMETHANE is available as an open-source Python software package that reads in input data, trains the model,

**Table 1** Key resource table

| Reagent or resource | Source | Identifier |
|---|---|---|
| Deposited data | | |
| 16S rRNA sequencing samples from He et al. [13] | European Nucleotide Archive (ENA) | ERP112481 |
| Metabolomics data from He et al. [13] | The curated gut microbiome-metabolome data resource (https://github.com/borenstein-lab/microbiome-metabolome-curated-data.wiki.git) [24] | HE-INFANTS-MFGM-2019 |
| 16S rRNA sequencing samples from Dawkins et al. [5] | Sequence Read Archive (SRA) | PRJNA772946 |
| Metabolomics data from Dawkins et al. [5] | [5] | 10.5281/zenodo.6473881 |
| Metagenomics sequencing data from Erawijantari et al. [16] | DNA Data Bank of Japan (DDBJ) Sequence Read Archive (DRA) | DRA007281, DRA008243, DRA006684 and DRA008156 |
| Metabolomics data from Erawijantari et al. [16] | The curated gut microbiome-metabolome data resource (https://github.com/borenstein-lab/microbiome-metabolome-curated-data.wiki.git) [24] | ERAWIJANTARI-GASTRIC-CANCER-2020 |
| Metagenomics sequencing data from Lloyd-Price et al. [14] | Sequence Read Archive (SRA) | PRJNA398089 |
| Metabolomics data from Lloyd-Price et al. [14] | Metabolomics Workbench (http://www.metabolomicsworkbench.org) | PR000639 |
| Metagenomics sequencing data from Franzosa et al. [15] | Sequence Read Archive (SRA) | PRJNA400072 |
| Metabolomics data from Franzosa et al. [15] | Metabolomics Workbench (http://www.metabolomicsworkbench.org) | PR000677 |
| Metagenomics sequencing data from Wang et al. [4] | European Bioinformatic Institute (EBI) database | PRJNA449784 |
| Metabolomics data from Wang et al. [4] | MetaboLights database (http://www.ebi.ac.uk/metabolights/) | MTBLS700 |
| Data to reproduce figures | This paper | 10.5281/zenodo.14064092 |
| Results of machine learning models run | This paper | 10.5281/zenodo.14064001 |
| Software and algorithms | | |
| MMETHANE model | https://github.com/gerberlab/mmethane [54] | 10.5281/zenodo.14058421 |
| Code to reproduce paper figures | https://github.com/gerberlab/mmethane-paper-results [55] | 10.5281/zenodo.14058419 |
| Metaphlan v3.0 | [26, 56] | N/A |
| DADA2 v1.20 | [25] | N/A |

and provides tools for visualization and interpretation of outputs.

Required and optional inputs and their file formats are as follows:

(1) Phylogenic tree *(optional, if default is used)*: a phylogenetic tree newick file (.nwk) or a table of pairwise phylogenetic distances between taxa (.csv,.tsv).
(2) Metabolite identification numbers *or* matrix of metabolite similarities/distances *(required)*: the software will automatically compute metabolite similarities (see below) given a (.csv,.tsv)-formatted table of metabolites and IDs for each metabolite measured. Identification numbers in any of the following formats are accepted: Human Metabolite DataBase (HMDB), KEGG, InChI, InChI Key, or PubChem. Alternatively, the user can provide a table (.csv,.tsv) of pre-computed pairwise distances or similarities between metabolites.
(3) Microbial counts/relative abundances *(required)*: a (.csv,.tsv)-formatted table, or a table provided by Metaphlan [26] or Dada2 [25].
(4) Metabolic measurements *(required)*: a (.csv,.tsv)-table of positive-valued metabolite levels.

MMETHANE supports automatic calculation of metabolite distances using five different molecular fingerprint methods:

(1) PubChem CACTVS (default): a fingerprint of 881-bits, in which each bit corresponds to the presence or absence of a chemical substructure. MMETHANE obtains these fingerprints using the PubChemPy API (v1.0.4) [27].
(2) Morgan: also referred to as extended-connectivity fingerprints (ECFP), this fingerprint is calculated by applying a variation of the Morgan algorithm, as described in [57]. MMETHANE obtains these fingerprints from RDKit (v2023.09.2) [58] using rdkit.Chem.AllChem.GetMorganFingerprintAsBitVect with bit-size set to the default of 2048 bits.
(3) Molecular Quantum Number (MQN) [29]: a vector of counts for 42 structural features falling into four categories: atoms, bond types, polarity, and topology. MMETHANE obtains these fingerprints using the function rdkit.Chem.rdMolDescriptiors.
(4) MinHashed atom-pair (MAP4) [59]**:** a 1024-bit fingerprint designed to be suitable for small and large molecules by combining elements of atom-pair and substructure fingerprints. MMETHANE obtains these fingerprints using map4 from the GitHub repository [28] with the default radius of 2 and default of 1024 dimensions.
(5) Infomax [30, 60]**:** a 300-bit fingerprint calculated using a pre-trained deep learning model. MMETHANE obtains the fingerprint and distances between the fingerprints by running the pre-trained 'gin_supervised_infomax' model [61] (dgllife v0.3.2, dgl v1.1.3) on bigraphs of all metabolites.

MMETHANE outputs cross-validated predictions for each sample as well as a PDF that contains the following:

(1) An English-language description of each rule.
(2) Heatmap visualizations of data used for model decisions.
(3) Visualizations of the phylogenetic relationships of the taxa in each detector.
(4) Visualizations of the metabolic structures of metabolites in each detector.

## MMETHANE model

### *Overview*

MMETHANE extends the MITRE/MDITRE models that we previously developed [22, 23], by adding detectors for metabolites. MMETHANE can thus integrate both metabolomic and microbial sequencing data, by learning predictive rules that contain both metabolite and taxa detectors. Note that MMETHANE does not implement the time-series analysis capabilities of MITRE/MDITRE, because sufficient datasets with paired longitudinal metabolite and microbial composition measurements were not available to validate this capability.

The MMETHANE model consists of a purpose-built, Bayesian, feed forward deep neural network (DNN), with the following layers: (i) feature aggregation, (ii) detector activation, (iii) detector selection and logical-AND rule activation, and (vi) rule selection and weighted-OR host status prediction (Fig. 1B). Prior biological knowledge is incorporated into the model using phylogenetic distances for microbial sequences and distances computed from molecular fingerprints for metabolites. Given the phylogenetic and metabolite distances, and training data consisting of paired metabolite and microbial sequencing data, the DNN learns a sparse set of detectors and rules that are then used to make predictions about the host label. As in our previous work [23], we use a Bayesian Variable Selection (BVS) procedure, for selecting the number of active detectors and rules. BVS provides a principled means to incorporate model sparsity and avoids bias that can occur with shrinkage methods, such as the lasso. For MMETHANE, as in our previous MDITRE model, we approximate the Bernoulli distribution with a continuous relaxation, the BinaryConcrete distribution, which approximates the Bernoulli distribution

that makes hard selections with a relaxed distribution that maintains differentiability throughout the model.

#### *Incorporating phylogenetic distances*

MMETHANE uses the same procedure for phylogenetic distance embedding as MDITRE [23]. Briefly, a symmetric $N_T \times N_T$ matrix of phylogenetic distances for $N_T$ input taxa is calculated from a reference phylogenetic tree. Embedding into Euclidean space is then performed using Principal Coordinate Analysis (PCoA), with the embedding dimensions $D_T$ automatically selected as the lowest dimension that does not result in a significant difference ($p < 0.05$, Kolmogorov–Smirnov (KS) test, SciPy v1.11.3) between the distributions of pre- and post-embedded distances. This yields an embedding matrix, $E^T \in \mathcal{R}^{N_T \times D_T}$ for the taxa. See Supplementary Table S5 for the embedding dimensions used for datasets in our compendium.

#### *Incorporating metabolite distances*

A symmetric $N_M \times N_M$ matrix of metabolite distances for $N_M$ metabolites is either provided by the user or calculated by the MMETHANE software package from provided metabolite identification numbers as described above. Distances are then embedded into Euclidean space using PCoA. We did not find that the KS-test yielded practically useful reductions in the dimensionality of the embedding space, as it had for phylogenetic distances. Thus, we chose the embedding dimension $D_M^f$ according to the number of components that explained 95% of variance for each fingerprinting method $f$. This procedure yields a set of embedding matrices, $E_f^M \in \mathcal{R}^{N_M \times D_M^f}$. See Supplementary Table S5 for the embedding dimensions used for datasets in our compendium.

#### *Sensitivity to embedding methods*

To assess model sensitivity to embedding methods, we compared MMETHANE's performance with PCoA to that with UMAP and tSNE (Supplementary Table 6). The Kruskal–Wallis test (Scipy v1.11.3) was used to determine any significant differences among results with different embedding methods. For the datasets in which results were statistically significant (3/6 datasets), the Mann–Whitney U test (Scipy v1.11.13) was used further differentiate between the three embedding methods. Overall, we found that neither of the other methods outperformed PCoA on any of the datasets.

#### *Model layers*

To define notation used in the model description below, let $K$ denote the total number of possible rules, $L$ the total number of possible taxa detectors per rule, and $J$ the total number of possible metabolite detectors per rule. As in our previous work, we set $K$, $J$, and $L$ to 10 [22, 23], which we found supports higher than utilized model capacity for the datasets analyzed.

(i) Feature aggregation

The first layer learns groups of features (taxa or metabolites) and outputs an aggregate of the features (sum of relative abundances for taxa, mean of averaged standardized levels for metabolites) for each of the learned detectors.

Grouping and aggregation for metabolites was modeled as follows. Let $\gamma_{jk}^M$ denote the learned center for metabolite detector $j$ in rule $k$. As in our previous work [23], we place a Multivariate Normal prior probability distribution with mean 0 and variance $10^4$ on detector centers.

Let $\kappa_{kj}^M$ denote the learned radius for detector $j$ in rule $k$. For metabolite detector radii, we use a Log-Normal prior calibrated with biological knowledge of metabolites. To obtain parameters for this distribution, first, a chemical taxonomy is assigned to each metabolite using Classy Fire [62]. Then, for each metabolite subclass $c$, we compute the pairwise distance, $q_{lp}$, between each pair of metabolites in that subclass using the metabolite embeddings $E^M$, and determine the median distance within each sub-class, $h_c^M$. The Log-Normal hyperparameters for the prior distribution on $\kappa_{kj}^M$ are then computed as the median and variance of the subclass distances, i.e.:

$$\kappa_{kj}^m \sim \text{LogNormal}(\text{median}_c(h_c), \text{variance}_c(h_c))$$

Note that sub-classes with less than three members were excluded from the analysis. We chose a nonparametric statistic for the first parameter, a median of medians, to provide robustness against outliers (i.e., very large sub-classes or smaller ones).

Each detector $j$ in rule $k$ determines whether to select metabolite $i$ according to the detector's center and radius, using a soft inclusion function:

$$u_{kji}^M = \text{sigmoid}\left(K_{kj}^M - \varepsilon_{kji}^M\right)/\tau_u$$

Here, $u_{kji}^M$ denotes the degree of selection, $\epsilon_{kji}^M$ denotes the Euclidean distance in the metabolite embedding space between the detector center and metabolite $i$, and $\tau_u$ is a temperature parameter that controls the sharpness of the selection. As in our previous work [23], we used a linear annealing schedule during training for the temperature parameter from $10^{-2}$ to $10^{-3}$.

Finally, for metabolites, the output of the layer is the mean of levels of selected metabolites in each detector:

$$a^M_{skj} = \frac{\sum_{i=1}^{N_M} u^m_{kji} X^m_{si}}{\sum_{i=1}^{N_M} u^m_{kji}}$$

Here, $X^M_{si}$ is the log transformed and standardized value of metabolite $i$ in subject $s$.

Grouping and aggregation of taxa was modeled for MMETHANE as in our previous work [23]. Briefly, taxa detectors learn centers and radii analogously to the metabolite detectors described. As we did for metabolites, we placed a Multivariate Normal prior on the centers and a Log-Normal prior on the radii. The Log-Normal prior mean and variance hyperparameters were set using median pairwise distances for each taxonomic family (with families with fewer than 3 taxa excluded). Finally, for taxa, the output of the layer is the sum of relative abundances of selected taxa in each detector:

$$a^T_{sk\ell} = \sum_{i=1}^{N_T} X^T_{si} u^T_{kli}$$

Here, $X^T_{si}$ is the relative abundance of taxa $i$ in subject $s$.

(ii) Detector activation

The second layer takes in the aggregated values for each feature group and learns a threshold of activation for each detector. The layer then outputs a soft binary activation for the detector, depending on whether the feature group aggregate is above or below the learned threshold.

We placed a uniform prior on the metabolite detector thresholds, $\eta^M_{kj}$, with end-points set based on the values of the metabolites in the data, $\boldsymbol{X}^m$:

$$\eta^M_{kj} \sim \text{Uniform}\left(\min(\boldsymbol{X}^M) - 0.01\boldsymbol{X}^M, \max(\boldsymbol{X}^M) + 0.01\boldsymbol{X}^M\right)$$

$\eta^T_{k\ell}$, the threshold for each taxa detector, is modeled by a uniform distribution from 0 to 1, as in [23].

The output of the layer is a soft activation modeled by a sigmoid centered around the learned thresholds for each metabolite or taxa detector:

$$g^M_{skj} = \text{sigmoid}(\left(a^M_{skj} - \eta^M_{kj}\right)/\tau^M_g$$
$$g^T_{sk\ell} = \text{sigmoid}(\left(a^T_{sk\ell} - \eta^T_{k\ell}\right)/\tau^T_g$$

The temperature parameters $\tau^M_g$ and $\tau^T_g$ control the sharpness of activation and are annealed during training to ensure final learned values of $g^M_{skj}$ and $g^T_{sk\ell}$ are close to 1 or 0. For taxa detectors, the temperature $\tau^T_g$ is annealed from $10^{-2}$ to $10^{-3}$ during training, as in [23]. The temperature for metabolite detectors, $\tau^M_g$, is annealed from 1 to 0.1 during training, because the range of values of $a^M_{skj}$ is typically 10 to 100 times larger than the range of values of $a^T_{sk\ell}$.

As in our previous work [23], we used sigmoid activations to encode "true" or "false" information in our logical rules. Sigmoid activations take on values between 0 and 1, and with the temperature annealing method used, take on values close to either 0 or 1 depending on whether the value of the input is greater or less than the rule threshold (or selection radius), mimicking discrete true or false decisions. We do not employ hard thresholds, because the model would then not be differentiable, which is needed for use of the standard deep learning libraries that we leverage for inference.

(iii) Detector selection and logical-AND rule activation

The third layer, takes in the outputs of the previous layer, the soft binary activations $\boldsymbol{g}_{sk} = \left[\boldsymbol{g}^m_{sk}, \boldsymbol{g}^t_{sk}\right]$ for all possible metabolite and taxa detectors, and selects a sparse set of detectors for the rule. The output of this layer, $r_{sk}$, is a soft binary activation for each rule $k$ and subject $s$, in which the value is approximately 1 when all its selected detectors are approximately 1, as described below.

For each rule k, we denote the set of $J+L$ detector selectors as $\boldsymbol{z}_k = \left[\boldsymbol{z}^M_k, \boldsymbol{z}^T_k\right]$. As in [23], we place Binary Concrete distribution priors on the selectors:

$$P\left(z^M_{k0}, \ldots, z^M_{kJ}\right) \sim \prod_j \text{BinaryConcrete}\left(z^M_{kj}; \alpha^M_z, \tau_z\right)$$
$$P\left(z^T_{k0}, \ldots, z^T_{kJ}\right) \sim \prod_j \text{BinaryConcrete}\left(z^T_{kj}; \alpha^T_z, \tau_z\right)$$

We set the location parameters, $\alpha^M_z$ and $\alpha^T_z$, to $1/J$ and $1/L$, respectively, to encourage sparsity, i.e., a prior assumption of expected values of 1 microbial and 1 metabolite detector per rule. The temperature parameter, $\tau_z$ is annealed from 1 to 0.1 during training, as in [23].

The output of the layer is a soft logical-AND function of the selected detector activations, as in our previous work [23]:

$$r_{sk} = \prod_j \left(1 - z_{kj}\left(1 - g_{skj}\right)\right)$$

(iv) Rule selection and weighted-OR host status prediction

In the final layer of the model, a sparse, interpretable ensemble of logical rules is combined to predict host status. Here, the model learns which of the $K$ rules to select for the prediction and weights for each selected rule. Analogously to the detector selectors, we placed Binary Concrete distribution priors on the rule selectors $\boldsymbol{q}$:

$$P(q_0, \ldots, q_K) \sim \prod\nolimits_j \text{BinaryConcrete}\left(q_k; \alpha_q, \tau_q\right)$$

As in our previous work, we set $\alpha_q$ to 1/K to encourage sparsity, i.e., a prior expectation of one rule, and $\tau_q$ was annealed from 1 to 0.1 during training. We placed a diffuse prior on the weights and biases for rules, $\beta_k$ and $\beta_0$ (a Normal distribution with mean 0 and variance $1 \times 10^4$).

The output of the layer for each subject $s$ is then the weighted linear combination of selected rule activations and the bias term:

$$Y_s = \text{logistic}\left(\sum_k q_k \beta_k r_{sk} + \beta_0\right)$$

Thus, the final prediction of host status is an ensemble of sparse, interpretable rules.

#### Inference

We employed maximum a posteriori (MAP) inference to learn the model parameters, using the Adam optimizer in PyTorch (v2.2.0). Initialization of the feature aggregation layer centers and radii was the same as in MDITRE [23]. Briefly, centers and radii were initialized by performing K-means clustering (using KMeans from scikit-learn v1.3.2) for the set of detectors. Detector activation thresholds were initialized as the average over all subjects of the feature groups determined by KMeans. The prediction layer weights and biases were initialized from a standard Normal distribution. The rule and detector selectors were initialized to 0.5, as in [23]. As in our previous work [23], learning rates were set to be 0.001 for all parameters, except for taxa detector thresholds, $\boldsymbol{\eta}^T$, which were set to be 0.0001. All temperature parameters ($\tau_u$, $\tau_g^M$, $\tau_g^T$, $\tau_z$, and $\tau_q$) were linearly annealed throughout training as described above. The model was trained for 5000 epochs for all datasets. Model convergence was defined as a decrease in training loss of less than 0.01 within the last 100 epochs; in all cases convergence was achieved within 5000 epochs.

MMETHANE uses an optimization-based inference method that in general can exhibit convergence to different local minima with different initializations. To mitigate this issue, by default the software runs the inference procedure ten times with different random seeds and reports the results for the initialization that achieves the lowest loss. We assessed the sensitivity of our inference procedure to initializations by analyzing learned detectors and rules across ten runs on all six datasets in our compendium (with each run consisting of ten seeds, and taking the result with the lowest loss as described above). Because the rules and detectors that MMETHANE learns have nested structures and can potentially encode the same/similar information with different structures (e.g., permutations of detectors or rules), straight-forward measures for associations of single features aren't clearly applicable. We instead used a bipartite matching metric, analogous to that employed in the clustering literature. Briefly, we compared runs by finding the optimal matches between detectors in runs, with Jaccard similarity as the measure of matching. Across the six datasets, we found an average Jaccard similarity of 0.52. This result indicates substantial consistency of features, with a greater than majority overlap between detector members (taxa or metabolites) across runs. However, because rules and detectors can vary from run to run, for exploratory analyses, we recommend that users compare results across multiple runs (we suggest ten) and focus on consistent findings for interpretation.

For reference, run-times on each data set in the compendium are reported in Supplementary Table 7. We note that the slowest run-time on the datasets analyzed was approximately 32 min (3.2 min per seed for 10 seeds) on an Apple M2 Max 12 core CPU, and used 4.8 GB RAM.

#### Semi-synthetic data generation

Semi-synthetic data was generated from real data using a parametric bootstrapping procedure, similar to that in our prior work [22, 23] for generating longitudinal taxa counts, but modified to generate metabolites as well, and without temporal information. We simulated data from the 67 control samples of Wang et al. [4], which consist of metabolite levels measured with HPLC–MS/MS and taxa counts measured with shotgun metagenomic sequencing.

The overall procedure for generating semisynthetic data was as follows:

(1) Sample a set of subjects from the control subjects of Wang et al. and split the set into "cases" and "controls."
(2) Sample a group (or groups) of metabolites and/or taxa to be perturbed.
(3) Calculate parameters for sampling the magnitude of synthetic perturbations, where the ranges of parameters were estimated from rules on real data.
(4) Generate data for "cases" and "controls" from the distributions determined in step 3.

Each of these steps is described in further detail below.

(1) Sampling sets of subjects

Subjects were uniformly sampled from the 67 control samples of Wang et al., with replacement if the number of subjects exceeded 67 in the simulation, and the set was then evenly split into "cases" and "controls." Note that subjects were sampled in a nested manner to assure comparability of results across different sample sizes, i.e., 300 subject cases were sub-sampled from 1000 subject cases,

128 subject cases were sub-sampled from 300 subject cases, etc.

(2) Sampling groups of metabolites or taxa to be perturbed

The procedure for sampling metabolite groups was as follows:

1. Select a metabolite $p$ uniformly at random from all the metabolites measured in the dataset, and obtain the metabolite $p$'s location in embedding space, $E_p^M$.
2. Sample a radius $\kappa$ from a Log-Normal distribution, with parameters set by the same procedure as described above for MMETHANE layer 1.
3. Calculate pairwise distances between all metabolites and $E_p^M$, and select the group of metabolites to be perturbed as all those within distance $\kappa$ from $E_p^M$.

To prevent the metabolite group from being unrealistically large, group sizes were restricted to a maximum of 15% of the total metabolites in the data; this was enforced via rejection sampling.

We followed the same procedure for sampling taxa groups to be perturbed as in our previous work [22, 23]. Briefly, clades were sampled uniformly from the tree with the restriction that clades contain a minimum of 5 members and a maximum of 30 members.

(3) Calculating perturbation magnitudes

Metabolites: To introduce a synthetic perturbation into the real metabolomics data, we sampled from a "control" distribution, Normal$(\mu_0^{\mathrm{M}}, \sigma_0^{\mathrm{M}})$, for a portion of subjects, and from a "case" distribution, Normal$(\mu_1^{\mathrm{M}}, \sigma_1^{\mathrm{M}})$, for the other subjects. We determined means of the distributions using the following procedure:

1. For each of the 6 datasets, determine the seed with the lowest loss for that dataset and obtain the set of MMETHANE rules for that seed. Over all the rules found in all the datasets, find the rule $k'$ with the largest regression coefficient in absolute value.
2. Calculate the fold-changes for all detectors $j$ in rule $k'$:

$$f_{k'j} = \frac{\frac{1}{|S_0|}\sum_{s\in S_0} g^M_{k'js}}{\frac{1}{|S_1|}\sum_{s\in S_1} g^M_{k'js}}$$

Here $S_0$ is the set of control subjects and $S_1$ the set of case subjects in the dataset that the rule was learned from, and $g^M_{k'js}$ is the average of the log-transformed and standardized levels of metabolites in rule $k'$, detector $j$.

1. Choose the detector $j'$ associated with the median fold-change, and calculate:

$$\mu_0^M = \frac{1}{|S_0|}\sum_{s\in S_0} g^M_{k'j's}$$

$$\mu_1^M = \frac{1}{|S_1|}\sum_{s\in S_1} g^M_{k'j's}$$

From this procedure, we found $\mu_0^M = -0.203$ and $\mu_1^M = 0.724$. We set $\sigma_0^M$ and $\sigma_1^M$ to 1.5, as in [22].

Taxa: We used the same values as in our previous work [22], $\mu_0^T = -6$, $\mu_1^T = -3$, and $\sigma_0^T = \sigma_0^M = 1.5$.

(4) Generating data

Scenario 1: 1 metabolite group perturbed

For each metabolite $i$ in the group to be perturbed, data was generated from a hierarchical model (top level modeling biological variation and the bottom level modeling measurement variability). Here, let $H$ denote the set of metabolites selected to be perturbed in the chosen group and $\boldsymbol{z}_s^M$ the standardized log-transformed values of all metabolites in subject $s$. To generate the semisynthetic data for each metabolite $m$, the following procedure was used:

1. Compute $p_{sm}$:
   - if m is in H (the set of perturbed metabolites), sample:

$$p_{sm} \sim \begin{cases} \text{Normal}\left(\mu_0^{\mathrm{M}}, \sigma_0^{\mathrm{M}}\right) \text{ if subject s is a } ''\text{control}'' \\ \text{Normal}\left(\mu_1^{\mathrm{M}}, \sigma_1^{\mathrm{M}}\right) \text{ if subject s is a } ''\text{case}'' \end{cases}$$

   - otherwise,

$$\text{set } p_{sm} = z_{sm}^M$$

2. Sample measurement error: $\epsilon_{sm} \sim \text{Normal}(0, \sigma^2_{\text{MEAS}})$. Here, $\sigma^2_{MEAS} = 0.024$, which we calculated from replicates from Dawkins et al. [5].
3. Un-transform the data back to its original space:

$$y'_{sm} = p_{sm} + \epsilon_{sm}$$
$$y^{M}_{sm} = e^{(y_{sm}\prime v_m + u_m)}$$

Here, $u_m$ and $v_m$ are the empirical mean and variance, respectively, of the untransformed levels of metabolite $m$ over all subjects.

Scenario 2: 2 metabolite groups perturbed

This scenario is similar to scenario 1, except we separate the subjects in the "control" group into those receiving a perturbation in the first metabolite group ("control 1"), and those receiving a perturbation in the second metabolite group ("control 2"). Here, let $m_1$ indicate the index of a metabolite in the first perturbation group $H_1$, and $m_2$ indicate the index of a metabolite in the second perturbation group $H_2$. Then, we sample values for metabolite $m_1$ as:

$$p_{sm_1} \sim \begin{cases} \text{Normal}\left(\mu_1^{\text{M}},\ \sigma_1^{\text{M}}\right) \text{ if subject s is a "case"} \\ \text{Normal}\left(2\mu_1^{\text{M}},\ \sigma_1^{\text{M}}\right) \text{ if subject s is in "control 1"} \\ \text{Normal}\left(\mu_0^{\text{M}},\ \sigma_0^{\text{M}}\right) \text{ if subject s is in "control 2"} \end{cases}$$

Similarly, for metabolite $m_2$, we sample:

$$p_{sm_2} \sim \begin{cases} \text{Normal}(\mu_1^{\text{M}}, \sigma_1^{\text{M}}) \text{ if subject s is a "case"} \\ \text{Normal}\left(\mu_0^{\text{M}}, \sigma_0^{\text{M}}\right) \text{ if subject s is in "control 1"} \\ \text{Normal}(2\mu_1^{\text{M}}, \sigma_1^{\text{M}}) \text{ if subject s is in "control 2"} \end{cases}$$

As in Scenario 1, if the metabolite is not in the group to be perturbed, $p_{sm}$ is set to its original value in the data.

Note that these parameter settings (i.e., alternately doubling the magnitude of the perturbation for subjects in either of the control subgroups) ensure that the expected total amount of all perturbations is consistent in subjects.

Scenario 3: 1 taxonomic clade perturbed

We follow the same procedure as in our prior work [22]. For completeness, we describe it here as well. Let $C$ denote the set of taxa selected to be perturbed in clade $c$ and $x^T_{so}$ the relative abundance of taxa $o$ for subject $s$.

1. For each subject $s$, sample a perturbation for the clade:

$$p'_{cs} \sim \begin{cases} \text{Normal}\left(\mu_0^{\text{T}}, \sigma_0^{\text{T}}\right) \text{ if subject s is a "control"} \\ \text{Normal}\left(\mu_1^{\text{T}}, \sigma_1^{\text{T}}\right) \text{ if subject s is a "case"} \end{cases}$$
$$p_{cs} = e^{p_{cs}\prime}$$

2. For each taxa compute $\gamma_{so}$:
   - If taxa $o$ is in $C$, $\gamma_{so}$ is the proportion of the sampled perturbation corresponding to the relative abundance of taxa $o$ in clade $c$ in the original data:

$$\gamma_{so} = p_{cs} \times \frac{w_{so}}{\sum_{i \in C} w_{si}}$$

   - Otherwise, $\gamma_{so}$ is set to the original relative abundance of taxa $o$:

$$\gamma_{so} = w_{so}$$

3. Add measurement noise in two stages:
   - First, sample from a truncated normal centered around $\gamma_{so}$ with standard deviation set to 30% of the mean ($\theta$=0.3)

$$\phi_{so} \sim \text{TruncatedNormal}\left(\gamma_{so}, (\theta\gamma_{so})^2, 0, 1\right)$$

   - Second, generate counts from a Dirichlet-Multinomial distribution:

$$y_s^T \sim \text{DMD}(\phi_s, \alpha, N)$$

Here, $\alpha$ is set to 286 as in [22], and $N$ is the total number of counts in subject $s$.

Scenario 4: 2 taxa clades perturbed

This scenario is analogous to Scenario 2, except using the taxa perturbation procedure described in Scenario 3.

Scenario 5: 1 taxa clade and 1 metabolite group perturbed

This scenario is analogous to Scenarios 2 and 4. We split controls into "control 1" and "control 2" groups. "Control 1" members then receive the taxa perturbation, "control 2" members receive the metabolite perturbation, and the "cases" receive both perturbations.

## Data pre-processing and filtering

### *Processing metabolomics data*

Metabolomics data was obtained from the original studies as feature intensities. These feature intensities were then filtered to remove metabolites in less than 15% of subjects, as in [5] and were then log-transformed and standardized.

### *Processing sequencing measurements*

Sequencing measurements were re-processed from FASTQ files with the same bioinformatic tools to ensure consistency. For datasets with 16S rRNA amplicon sequencing, the raw data was reprocessed with DADA2 v1.20 [25] and the resulting amplicon sequencing variants were phylogenetically placed on a reference tree as described in [22]. For datasets with metagenomics

sequencing, the raw sequences were processed with Metaphlan v3.0 [26, 56]. For both types of data, filtering was done as in [22]: taxa with counts below 10 in less than 10% of subjects were removed.

#### *Sensitivity to filtering thresholds*

To assess the sensitivity of results to the specific filtering thresholds used on the percent of subjects in which a metabolite or taxa needed to be detectable, we ran MMETHANE and all other comparator methods on the datasets in the compendium with thresholds of 10%, 15%, and 20% for both taxa and metabolites (nine different filtering combinations over six datasets and using 5 different machine learning combinations). Out of the 270 combinations, only ~ 5% (15 cases) were significantly different for at least one of the filtering thresholds, with no threshold producing consistently better results. Overall, these analyses demonstrate insensitivity to the filtering thresholds chosen.

#### *Processing datasets with multiple timepoints*

Two datasets in our compendium, He et al. and Lloyd-Price et al., sampled multiple timepoints. However, the temporal information available in both datasets was insufficient to be useful for predictive modeling, as only the first two timepoints of He et al. showed any significant difference between groups and the time points sampled in Lloyd-Price et al. were substantially inconsistent across subjects. Thus, for both datasets, we selected one timepoint to use for analysis. For both He et al. and Lloyd Price et al., we used the timepoint that resulted in the largest sample size for each participant. For He et al., this was the timepoint at 2 months of life, and for Lloyd-Price et al., this was the first timepoint sampled from each participant.

#### Comparator machine learning methods

We benchmarked our model against lasso logistic regression (LR), random forest (RFs), adaptive boosting (AdaBoost), and feed forward neural network (FFNNs). For the ensemble benchmarking methods (RFs AdaBoost), data processing and filtering were performed identically as described MMETHANE. For LR and FFNNs, taxa relative abundances were transformed using the centered-log ratio transformation and then standardized. All methods used nested cross-validation, with the AUC as the metric, to tune hyperparameters. Nested fivefold cross-validation was performed for all datasets, except the CDI dataset, on which nested leave-one-out cross-validation was performed due to the small sample size.

All comparator methods except FFNNs were implemented with Scikit-Learn (v1.3.2). LR parameter tuning was performed using LogisticRegressionCV, with the input settings and hyperparameter grid search performed as in [5]. RFs and AdaBoost parameter tuning was performed using GridSearchCV. For RFs, the hyperparameter grid to be searched over was set as in [5]. For AdaBoost, the hyperparameter grid was searched over 50 or 100 estimators and learning rates from $10^{-2}$ to $10^{2}$.

FFNNs were implemented in PyTorch, (v2.2.0) as a fully-connected, feed forward network with 3 layers. The number of nodes in each layer was scaled to the number of input features in the dataset using the standard procedure for selecting layer sizes outlined in [63]. Supplementary Table 8 provides the layer sizes used for each dataset in the compendium. Drop out was used to prevent overfitting, with a rate of $p = 0.2$ selected. Models were run for 2000 epochs with a learning-rate of 0.001, dropout 0.1, and weight decay of 0.01. Models were assumed to have converged when the decrease in training loss over a window of 100 epochs was less than 0.01.

#### Determining modalities of predictors

For MMETHANE, predictor modality was directly determined from the model, using the rule set from the seed corresponding to the lowest total loss. For LR, we calculated the median odds and 95% interval of each feature over 10 seeds and 5 cross-validation folds within each seed, and retained every feature whose 95% odds-interval did not contain 0, as in [5]; modalities of retained features were then counted. We did the same for RFs and AdaBoost, but with Gini importance rather than odds. For FNNs, we calculated feature importances as the average integrated gradients using Captum (v0.7.0) [64], over ten seeds. Because no feature importances for the FNN were 0, we used the top 10 features for subsequent analyses.

### Quantification and statistical analysis

The Mann Whitney U test from the Stats module of SciPy (v.1.11.3) was used to compare benchmarking results of methods on real and semi-synthetic data, with significance defined as $p < 0.05$. All other details of statistical analyses and software can be found in the Methods section above, but are briefly summarized here. Calculation of the embedding dimension for taxa was performed using the KS test ($p = 0.05$) from SciPy Stats (v.1.11.3). All machine learning models were run in Python (v3.11) using PyTorch (v2.2.0) and/or Scikit-Learn (v.1.3.2).

## Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40168-025-02270-z.

Supplementary Material 1: Supplementary Figure S1. The PubChem CACTVS fingerprint yielded the most robust predictive performance when used as a molecular similarity measure for MMETHANE. Five-fold cross-validated AUC scores for prediction of host status on the six datasets in the compendium are shown, using five different molecular similarity measures (PubChem CACTVS, Morgan, MAP4, MQN and InfoMax). Box plots indicate medians and 95% intervals for runs over ten random seeds. Yellow diamonds indicate the top score or scores (if multiple scores were not significantly different from the top score, $p > 0.05$). Supplementary Table S1. Compendium of paired microbial composition and metabolomics datasets. NMR = nuclear magnetic resonance. CDI = *Clostridioides difficile* infection. HPLC–MS/MS = high-performance liquid chromatography-tandem mass spectrometry. IBD = inflammatory bowel disease. LC–MS = liquid chromatography-mass spectrometry. ESRD = End-stage renal disease. Supplementary Table S2. Chemical fingerprints assessed and rationale for selection. Supplementary Table S3. Benchmarking results for comparator methods using aggregated taxa or metabolites based on pre-defined groupings as inputs. Benchmarking methods were evaluated on aggregated taxonomic family or metabolic class-level predictors. Cross-validated AUC values with ranges are reported as described in the main text and Methods. Results from aggregated predictors and non-aggregated predictors for each dataset and each benchmark method were compared with Mann–Whitney U tests, with bolded scores indicating significantly better results. Overall, models given non-aggregated data performed better than models given aggregated data. Supplementary Table S4. Modality (metabolite or taxa) of predictors found when both metabolomics and taxa abundance data were given to models as inputs. LR = lasso logistic regression, RF = random forests, FFNN = feedforward neural networks. For LR, features were retained if their 95% cross-validated odds interval over 10 seeds did not contain 0. For RFs and AdaBoost, features were retained if their 95% cross-validated Gini importance over 10 seeds did not contain 0. For FNNs, the top 10 features were retained, ranked based on the mean of the integrated gradient of each feature over all subjects and 10 seeds. Supplementary Table S5. Embedding dimensions for taxa and metabolites on the datasets in the compendium. Note that the embedding dimensions for metabolites and taxa were approximately the same within and across datasets, with the exception of He et al. and Dawkins et al., which had lower embedding dimensions for taxa, because these datasets used 16S rRNA phylotyping in contrast to metagenomic sequencing used in the other datasets. Supplementary Table S6. Benchmarking results for MMETHANE using different embedding methods for detectors. We compared MMETHANE's performance on the data compendium using UMAP, PCoA, and tSNE for embeddings of metabolites and taxa. In three of the six datasets, there were no statistically significant difference between results. Cross-validated AUC values with ranges are reported as described in the main text and Methods. The Kruskal–Wallis test was used to determine any significant differences among results with different embedding methods. For the datasets in which results were statistically significant, the Mann–Whitney U test was used to further differentiate between embedding methods. Bolded scores indicate significantly better results. In three of the datasets, there were no significant differences between the embedding methods. In Wang et al. and Dawkins et al., PCoA performed significantly better than UMAP and tSNE, while in Erawijantari et al., PCoA and tSNE performed significantly better than UMAP. Supplementary Table S7. Median run time per seed in minutes on an Apple M2 Max 12 core CPU. Supplementary Table S8. Feedforward Neural Network (FFNN) layer sizes for each dataset in the compendium. Note that the input layer size reflects the total number of taxa and metabolites in the dataset.

**Acknowledgements**
We would like to thank Liat Shenov and Travis Gibson for helpful feedback and discussions about the MMETHANE model and datasets analyzed.

**Authors' contributions**
Conceptualization, G.K.G. and J.D.; Methodology, G.K.G. and J.D.; Software, J.D.; Validation, J.D.; Formal Analysis, J.D.; Investigation, J.D.; Resources, G.K.G; Writing, G.K.G. and J.D.; Visualization, J.D..; Supervision, G.K.G.; Funding Acquisition, G.K.G. All authors read and approved the final manuscript.

**Funding**
This work was supported by NSF MTM2 2025512, NIH NIGMS R01GM130777, NIH NIGMS R35GM149270, The Massachusetts Life Sciences Center, and the Brigham and Women's Hospital President's Scholar Award.

**Data availability**
The MMETHANE model is available at [https://github.com/gerberlab/mmethane](https://secure-web.cisco.com/1Vud2rlAm-2ctbW-qcde-zueau6n9HtrkwEUQGv8iPbS2G51WacbE2BdqynzJv-poRWHd-o3O9yt8Ybuf0zXKsv8XywNVaLofYAVHSwAdhFxDrgoYi8vyAwftwmNKkrSRxQdj-WhNJxXza7zDz084HwgpJvZKXSoZEEJ91XT8KXXxGSK_NzIHa0HO7GnonL-N0WBcOraDcK5wETBwKqq9UwWsNwArgz41P4OKmzZTyNdM3E_SS4tTNcyVawbnzo_rGh5AwS7RRYlNoHUhSfq02neXN_-5rPWymF3dBtz4lvrGXZtycvFZVbn0RTuo1NAv/https%3A%2F%2Fgithub.com%2Fgerberlab%2Fmmethane) [ [54](.) ]. Code to reproduce all analyses and figures in this project, as well as the semi-synthetic and real datasets used to produce results is available at [https://github.com/gerberlab/mmethane-paper-results](https://github.com/gerberlab/mmethane-paper-results) [ [55](.) ]. Data needed to reproduce all figures is available at https://doi.org/10.5281/zenodo.14064092.

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

**Competing interests**
G.K.G. holds shares of NVIDIA in a retirement account; he has no other financial interest in the company and the company had no involvement in this work. J.D. declares no competing interests.

Received: 18 December 2024 Accepted: 22 September 2025
Published online: 08 December 2025

**References**

1. Levy M, Blacher E, Elinav E. Microbiome, metabolites and host immunity. Curr Opin Microbiol. 2017;35:8–15.
2. Nicholson JK, et al. Host-gut microbiota metabolic interactions. Science. 2012;336(6086):1262–7.
3. Afzaal M, et al. Human gut microbiota in health and disease: unveiling the relationship. Front Microbiol. 2022. https://doi.org/10.3389/fmicb.2022.999001.
4. Wang X, et al. Aberrant gut microbiota alters host metabolome and impacts renal failure in humans and rodents. Gut. 2020;69(12):2131–42.
5. Dawkins JJ, et al. Gut metabolites predict *Clostridioides difficile* recurrence. Microbiome. 2022;10(1):87.
6. Wu WK, et al. Measurement of gut microbial metabolites in cardiometabolic health and translational research. Rapid Commun Mass Spectrom. 2020. https://doi.org/10.1002/rcm.8537.
7. Liu H, et al. Butyrate: a double-edged sword for health? Adv Nutr. 2018;9(1):21–9.
8. Chen J, Vitetta L. Butyrate in inflammatory bowel disease therapy. Gastroenterology. 2020;158(5):1511.
9. Pike CM, Theriot CM. Mechanisms of colonization resistance against *Clostridioides difficile*. J Infect Dis. 2021;223(12 Suppl 2):S194–200.
10. Pavao A, et al. Reconsidering the in vivo functions of Clostridial Stickland amino acid fermentations. Anaerobe. 2022;76:102600.

11. Schrimpe-Rutledge AC, et al. Untargeted metabolomics strategies—challenges and emerging directions. J Am Soc Mass Spectrom. 2016;27(12):1897–905.
12. Vellido A. The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural Comput Appl. 2020;32(24):18069–83.
13. He X, et al. Fecal microbiome and metabolome of infants fed bovine MFGM supplemented formula or standard formula with breast-fed infants as reference: a randomized controlled trial. Sci Rep. 2019;9(1):11589.
14. Lloyd-Price J, et al. Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. Nature. 2019;569(7758):655–62.
15. Franzosa EA, et al. Gut microbiome structure and metabolic activity in inflammatory bowel disease. Nat Microbiol. 2019;4(2):293–305.
16. Erawijantari PP, et al. Influence of gastrectomy for gastric cancer treatment on faecal microbiome and metabolome profiles. Gut. 2020;69(8):1404–15.
17. Larose, D.T., Data mining and predictive analytics. New York: John Wiley & Sons; 2015.
18. Broadhurst DI, Kell DB. Statistical strategies for avoiding false discoveries in metabolomics and related experiments. Metabolomics. 2006;2(4):171–96.
19. Yachida S, et al. Metagenomic and metabolomic analyses reveal distinct stage-specific phenotypes of the gut microbiota in colorectal cancer. Nat Med. 2019;25(6):968–76.
20. Sinha R, et al. Fecal microbiota, fecal metabolome, and colorectal cancer interrelations. PLoS ONE. 2016;11(3):e0152126.
21. Jiang T, Gradus JL, Rosellini AJ. Supervised machine learning: a brief primer. Behav Ther. 2020;51(5):675–87.
22. Bogart E, Creswell R, Gerber GK. MITRE: inferring features from microbiota time-series data linked to host status. Genome Biol. 2019;20(1):186.
23. Maringanti VS, Bucci V, Gerber GK. MDITRE: scalable and interpretable machine learning for predicting host status from temporal microbiome dynamics. mSystems. 2022;7(5):e0013222.
24. Muller E, Algavi YM, Borenstein E. The gut microbiome-metabolome dataset collection: a curated resource for integrative meta-analysis. NPJ Biofilms Microbiomes. 2022;8(1):79.
25. Callahan BJ, et al. DADA2: high-resolution sample inference from Illumina amplicon data. Nat Methods. 2016;13(7):581–3.
26. Truong DT, et al. Microbial strain-level population structure and genetic diversity from metagenomes. Genome Res. 2017;27(4):626–38.
27. PubChem substructure fingerprint V1.3. 2009. https://ftp.ncbi.nlm.nih.gov/pubchem/specifications/pubchem_fingerprints.txt.
28. map4. 2021, GitHub: GitHub repository. https://github.com/reymond-group/map4.
29. Awale M, van Deursen R, Reymond JL. MQN-mapplet: visualization of chemical space with interactive maps of DrugBank, ChEMBL, PubChem, GDB-11, and GDB-13. J Chem Inf Model. 2013;53(2):509–18.
30. Vellickovic, P., et al., Deep Graph Infomax. 2019. https://doi.org/10.48550/arXiv.1809.10341.
31. Zhou YH, Gallins P. A review and tutorial of machine learning methods for microbiome host trait prediction. Front Genet. 2019;10:579.
32. Ditzler G, Polikar R, Rosen G. Multi-layer and recursive neural networks for metagenomic classification. IEEE Trans Nanobioscience. 2015;14(6):608–16.
33. Nakano Y, Suzuki N, Kuwata F. Predicting oral malodour based on the microbiota in saliva samples using a deep learning approach. BMC Oral Health. 2018;18(1):128.
34. Zeevi D, et al. Personalized nutrition by prediction of glycemic responses. Cell. 2015;163(5):1079–94.
35. Thaiss CA, et al. Persistent microbiome alterations modulate the rate of post-dieting weight regain. Nature. 2016;540(7634):544–51.
36. Moitinho-Silva L, et al. Predicting the HMA-LMA status in marine sponges by machine learning. Front Microbiol. 2017;8:752.
37. Wu H, et al. Metagenomics biomarkers selected for prediction of three different diseases in Chinese population. Biomed Res Int. 2018;2018:2936257.
38. Du M, Liu N, Hu X. Techniques for interpretable machine learning. Commun ACM. 2019;63(1):68–77.
39. Liu J, Nussinov R. Allostery: an overview of its history, concepts, methods, and applications. PLoS Comput Biol. 2016;12(6):e1004966.
40. Singh V, et al. Butyrate producers, "The Sentinel of Gut": Their intestinal significance with and beyond butyrate, and prospective use as microbial therapeutics. Front Microbiol. 2022;13:1103836.
41. Bergey, D.H., et al., Bergey's manual of systematic bacteriology. Vol. 3, The firmicutes. 2nd ed. 2009, New York: Springer.
42. Portincasa P, et al. Gut microbiota and short chain fatty acids: implications in glucose homeostasis. Int J Mol Sci. 2022. https://doi.org/10.3390/ijms23031105.
43. Kwon YH, et al. Modulation of gut microbiota composition by serotonin signaling influences intestinal immune response and susceptibility to colitis. Cell Mol Gastroenterol Hepatol. 2019;7(4):709–28.
44. Haq S, et al. Disruption of autophagy by increased 5-HT alters gut microbiota and enhances susceptibility to experimental colitis and Crohn's disease. Sci Adv. 2021;7(45):eabi6442.
45. Shkoporov AN, et al. *Ruthenibacterium lactatiformans* gen. nov., sp. nov., an anaerobic, lactate-producing member of the family Ruminococcaceae isolated from human faeces. Int J Syst Evol Microbiol. 2016;66(8):3041–9.
46. Gossling J, Moore WEC. *Gemmiger formicilis*, n.gen., n.sp., an anaerobic budding bacterium from intestines. Int J Syst Evol Microbiol. 1975;25(2):202–7.
47. Russell WR, et al. Major phenylpropanoid-derived metabolites in the human gut can arise from microbial fermentation of protein. Mol Nutr Food Res. 2013;57(3):523–35.
48. Roager HM, Licht TR. Microbial tryptophan catabolites in health and disease. Nat Commun. 2018;9(1):3294.
49. Ren W, et al. Dietary arginine supplementation of mice alters the microbial population and activates intestinal innate immunity. J Nutr. 2014;144(6):988–95.
50. Baier J, et al. Arginase impedes the resolution of colitis by altering the microbiome and metabolome. J Clin Invest. 2020. https://doi.org/10.1172/JCI126923.
51. Wu M, et al. Arginine accelerates intestinal health through cytokines and intestinal microbiota. Int Immunopharmacol. 2020;81:106029.
52. Nüse B, et al. L-arginine metabolism as pivotal interface of mutual host-microbe interactions in the gut. Gut Microbes. 2023;15(1):2222961.
53. Liu Y, et al. *Clostridium sporogenes* uses reductive Stickland metabolism in the gut to generate ATP and produce circulating metabolites. Nat Microbiol. 2022;7(5):695–706.
54. Dawkins, J.G., G. MMETHANE [Computer Software]. 2024; Available from: https://github.com/gerberlab/mmethane.
55. Dawkins, J.G., G. MMETHANE Paper Results (Version 0.0) [Computer Software]. 2024; Available from: https://github.com/gerberlab/mmethane-paper-results.
56. Beghini F, et al. Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with bioBakery 3. Elife. 2021;10:e65088.
57. Rogers D, Hahn M. Extended-connectivity fingerprints. J Chem Inf Model. 2010;50:742–54.
58. RDKit: Open-source cheminformatics. https://github.com/rdkit/rdkit.
59. Capecchi A, Probst D, Reymond JL. One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. J Cheminform. 2020;12(1):43.
60. Li M, et al. DGL-lifesci: an open-source toolkit for deep learning on graphs in life science. ACS Omega. 2021;6(41):27233–8.
61. dgl-lifesci. 2023, GitHub: GitHub repository. https://github.com/awslabs/dgl-lifesci.
62. Djoumbou Feunang Y, Eisner R, Knox C. et al. ClassyFire: automated chemical classification with a comprehensive, computable taxonomy. J Cheminform. 2016;8:61. https://doi.org/10.1186/s13321-016-0174-y.
63. Stathakis D. How many hidden layers and nodes? Int J Remote Sens. 2009;30(8):2133–47.
64. Sundararajan, M., A. Taly, and Q. Yan. Axiomatic Attribution for Deep Networks. in International Conference on Machine Learning. 2017.


## Publisher's Note