DeepCoDA: personalized interpretability for compositional health data

Thomas P. Quinn [*1] **Dang Nguyen** [*1] **Santu Rana** [1] **Sunil Gupta** [1] **Svetha Venkatesh** [1]

Abstract

Interpretability allows the domain-expert to directly evaluate the model's relevance and reliability, a practice that offers assurance and builds trust. In the healthcare setting, interpretable models should implicate relevant biological mechanisms independent of technical factors like data pre-processing. We define *personalized interpretability* as a measure of sample-specific feature attribution, and view it as a minimum requirement for a precision health model to justify its conclusions. Some health data, especially those generated by high-throughput sequencing experiments, have nuances that compromise precision health models and their interpretation. These data are *compositional*, meaning that each feature is conditionally dependent on all other features. We propose the **Deep Co**mpositional **D**ata **A**nalysis **(DeepCoDA)** framework to extend precision health modelling to high-dimensional compositional data, and to provide personalized interpretability through patient-specific weights. Our architecture maintains state-of-the-art performance across 25 real-world data sets, all while producing interpretations that are both personalized and fully coherent for compositional data.

1. Introduction

Interpretability is pivotal for the adoption and use of predictive deep models in healthcare. As one physician noted, "Without [English-language] explanations, it is obviously unreasonable for the physician to rely on such programs; ultimately, a program, like any consultant, must justify its conclusions to the physician responsible for the patient's care" (Schwartz et al., 1987). Interpretability allows the domain-expert to directly evaluate the model's relevance and reliability, a practice that offers assurance and builds trust. In the healthcare setting, interpretable models should implicate relevant biological mechanisms independent of technical factors like data pre-processing. Precision health research aims to target disease prevention and health promotion to individual patients (Ashley, 2016). Although this field has benefited tremendously from the availability of clinical data, its mission requires new models that offer interpretability at the level of the individual patient. We define *personalized interpretability* as a measure of sample-specific feature attribution, and view it as a minimum requirement for a precision health model to justify its conclusions.

The *attention mechanism* offers one approach to personalized interpretability (Bahdanau et al., 2016), and has been used recently to study cancer heterogeneity (Beykikhoshk et al., 2020). The *self-explaining neural network* (SENN), may likewise enable precision health through its formulation $y = \theta(\mathbf{x})^\top \mathbf{x}$ (Alvarez-Melis and Jaakkola, 2018). This equation is well-suited to personalized interpretability for 3 reasons: (a) the input features x_j are clearly anchored to the observed empirical measurements; (b) each parameter $\theta(\mathbf{x})_j$ estimates the quantitative contribution of its corresponding feature x_j to the predicted value; and (c) the aggregation of feature specific terms $\theta(\mathbf{x})_j x_j$ is additive, allowing for a feature-by-feature interpretation of impact. However, self-explanation has not yet been tailored to high-throughput health biomarker data, which exist not as absolute measurements but rather multivariate compositions.

Much of the health data generated by high-throughput sequencing experiments have nuances that compromise precision health models and their interpretation. These data are compositional, meaning that they arise from an inexhaustive sampling procedure in which each feature is conditionally dependent on all other features (Gloor et al., 2017; Quinn et al., 2018; Calle, 2019). Compositional data can be defined as the proportions of $\mathbf{x}^*$ (Aitchison, 1986)

$$\mathbf{x} = \frac{[x_1^*, ..., x_D^*]}{\sum_j^D x_j^*} \tag{1}$$

where $\mathbf{x}$ is a composition with D parts. By considering this equation carefully, one can gain an intuition for why compositional data are so difficult to analyze. For example, consider the 3-part vector $\mathbf{x}^* = [a^*, b^*, c^*]$ of *absolute* abundances and its corresponding composition $\mathbf{x} = [a, b, c]$ of *relative* abundances. When c^* increases, c will also in-

[*] Equal contribution [1] Applied Artificial Intelligence Institute (A^{2}I^2), Deakin University, Geelong, Australia. Correspondence to: Thomas P. Quinn <contacttomquinn@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

crease (as expected). However, a and b must also decrease because the sum of $\mathbf{x}$ is fixed! As such, the value of any one part x_j depends on all other parts. For compositional data, common measures of association (Pearson, 1896; Lovell et al., 2015), distance (Aitchison et al., 2000), and feature attribution (Boogaart and Tolosana-Delgado, 2013a) can be misleading. Likewise, both supervised learning (Rivera-Pinto et al., 2018; Tolosana Delgado et al., 2019) and unsupervised learning (Avalos et al., 2018; Martín-Fernández et al., 2019) require an innovative approach.

In practice, analysts use normalization in an attempt to make the data absolute, but these rely on untestable assumptions (e.g., that the majority of features remain unchanged (Robinson and Oshlack, 2010; Anders and Huber, 2010)). Two manuscripts, both studying bacteria-metabolite associations, avoided normalization by having the hidden layers compute on log-ratio transformed data, either implicitly (through an inverse transform of the output layer (Morton et al., 2019)) or explicitly (through a transform of the input layer (Le et al., 2019)). The latter model sought interpretability through heavy regularization and a non-negative weights constraint, but its interpretation would still depend on a normalizing assumption even if its performance does not (c.f., (Erb and Notredame, 2016)). *We are unaware of any neural network architecture designed specifically for the personalized interpretation of compositional data.*

We propose a normalization-free neural network architecture called **DeepCoDA** to provide personalized interpretability for compositional data. We overcome 3 key challenges through the use of an end-to-end neural network:

- **A model should select the best log-ratio transformation automatically.** It is necessary to transform the feature space, but there are many ways to do this. We propose a *log-bottleneck module* to learn useful log-contrasts from the training data. It works by passing log-transformed data through a hidden layer with a single node such that the layer weights sum to 0. The node thus becomes a simple log-contrast, and B modules can be stacked in parallel to get B log-contrasts.
- **A model should have linear interpretability.** The use of non-linear transformations will often improve the predictive performance of a model, but this can come at the expense of interpretability. We use a *self-explanation module* to introduce linear interpretability, which has the form $y = \theta(\mathbf{x})^\top \mathbf{x}$ (Alvarez-Melis and Jaakkola, 2018). Each output is predicted by a linear combination of the input, just like a linear regression.
- **A model should have personalized interpretability.** Many models can estimate feature importance for the whole population, but precision health requires an estimation of importance for the individual patient. This is also achieved by the *self-explanation module* (mentioned above), which computes patient-specific weights as a function of the input.

Biological scientists routinely study the gut microbiome as a biomarker for disease prediction and surveillance (e.g., inflammatory bowel disease and cancer (Duvallet et al., 2017)). Microbiome data are compositional, and thus inherit the data nuances described above. We validate our **DeepCoDA** framework on 25 separate microbiome and similar data sets. Using a cohort of 13 data sets, we find a set of hyper-parameters that work generally well, and verify them on 12 unseen data sets. This two-step procedure allows us to recommend general hyper-parameters for real-world microbiome data.

Our framework solves an open problem in biology: how to extend personalized interpretability to compositional health data, including RNA-Seq, metagenomics, and metabolomics. The novelty lies in using neural networks to learn predictive log-contrasts, and coupling them with self-explanation to provide personalized interpretability through simple algebraic expressions. This combined architecture results in interpretations that are both personalized and fully coherent for compositional data, all while maintaining state-of-the-art performance.

2. Related Background

Compositional data analysis: The field of compositional data analysis (CoDA) emerged in the 1980s (Aitchison, 1986). Most often, CoDA uses an internal reference to transform the raw variables into a set of log-ratios for which the denominator is the reference. Log-ratio models are implicitly normalization-free because the normalization factor cancels by the ratio. Popular references include the per-sample geometric mean (*centered log-ratio transform*), a single feature of importance (*additive log-ratio transform*), or all features (*pairwise log-ratio transform*) (Aitchison, 1986). We include the centered log-ratio as part of the baseline used to benchmark our model.

Log-contrast models: Aitchison and Bacon-Shone (1984) proposed log-contrast models for the analysis of compositional data. According to the authors, log-contrasts are useful when an analyst wants to study the change of some parts of a composition while holding fixed some other parts. Like log-ratio models, log-contrast models are normalization-free because the normalization factor cancels. A subset of log-contrasts, called *balances*, have gained some popularity in microbiome research (Morton et al., 2017; Washburne et al., 2017; Silverman et al., 2017). Recent work has proposed data-driven heuristics, including *forward-selection* (Rivera-Pinto et al., 2018) and *cluster analysis* (Quinn and Erb, 2020), to identify predictive balances. These approaches take a statistical learning perspective that may not generalize to non-linear multivariable regression. We include balances as part of the baseline used to benchmark our model.

Parallel blocks: Tsang et al. (2018) proposed the Neural Interaction Transparency (NIT) architecture to discover statistical interactions among variables. This architecture uses multiple multi-layer perceptrons stacked as parallel *blocks*, where incoming connections define *feature sets*. Our proposed network also uses the idea that parallel blocks can define interpretable feature sets. For NIT, the authors interpret each feature set as a statistical interaction. They learn each feature set in parallel, and the output is predicted by an additive combination of the feature sets. For **DeepCoDA**, we interpret each feature set as a single log-contrast. We learn these log-contrasts in parallel, and the output is also predicted by an additive combination of the log-contrasts.

Attention: Bahdanau et al. (2016) used an attention mechanism for neural machine translation. Within a recurrent neural network, they compute a context vector for each time state i as the weighted sum of the encoded input. The weights are determined by an attention vector α as a function of the latent variables themselves. One could interpret the attention vector weights as a measure of feature importance, and thus view attention as an estimate of sample-specific importance. As such, attention is highly relevant to precision health, which aims to tailor medical care to the individual patient. Beykikhoshk et al. (2020) used attention to classify gene expression signatures, and analyzed the attention vectors directly to study disease heterogeneity.

Self-explaining neural networks: Alvarez-Melis and Jaakkola (2018) proposed the self-explaining neural network as a highly interpretable architecture that acts like a simple linear model locally, but not globally. They describe a linear model whose coefficients depend on the input itself, having the form $f(\mathbf{x}) = \theta(\mathbf{x})^\top \mathbf{x}$. When θ is a neural network, this function can learn complex non-linear functions while still having the interpretability of a linear model. The authors extend this function beyond input features to latent features, and propose a regularization penalty to force the model to act locally linear. In our work, we use the form $f(\mathbf{z}) = \theta(\mathbf{z})^{\mathrm{T}}\mathbf{z}$ to learn sample-specific weights for each log-contrast in $\mathbf{z}$. This is similar to computing an attention vector, but we use self-explanation because it is algebraically and conceptually simpler.

3. The Proposed Framework

3.1. Network architecture

Figure 1 provides a visual overview of our proposed neural network architecture for compositional data analysis. It contains 2 distinct modules.

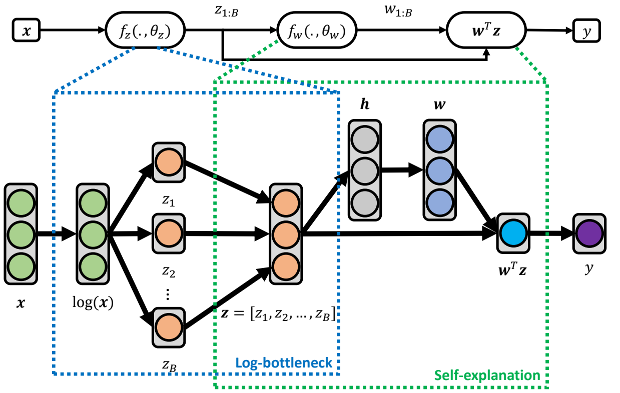

Figure 1. The network architecture has two distinct modules. The first is the *log-bottleneck module* which passes the log-transformed data through a bottleneck layer of a single hidden node such that the layer weights sum to 0. This hidden node therefore describes a single log-contrast, and B modules can be stacked in parallel to get B log-contrasts. The second is the *self-explanation module* which introduces both linear interpretability and personalized interpretability.

3.1.1. The log-bottleneck module

$$\mathbf{z}_i = f_z(\mathbf{x}_i; \theta_z) = [z_{i1}, \ldots, z_{iB}] \tag{2}$$

The first module is the *log-bottleneck*. The function f_z takes a composition of $d = 1 \ldots D$ features as input and returns $b = 1 \ldots B$ log-contrasts, parameterized by θ_z. This module is described in detail in Section 3.2.

3.1.2. The self-explanation module

$$\mathbf{w}_i = f_w(\mathbf{z}_i; \theta_w) = [w_{i1}, \ldots, w_{iB}] \tag{3}$$

$$y_i = f_a(\mathbf{w}_i, \mathbf{z}_i; \theta_a) = \mathbf{w}_i^{\mathrm{T}} \mathbf{z}_i \tag{4}$$

The second module is the *self-explanation*. The function f_w takes a vector of $b = 1 \ldots B$ log-contrasts as input and returns $b = 1 \ldots B$ weights, parameterized by θ_w. We define f_w as a neural network with a single hidden layer of 16 nodes and ReLu activation. The function f_a is a simple dot product between the input-specific weight vector $\mathbf{w}_i$ and the log-contrasts $\mathbf{z}_i$, so θ_a is a null set. Thus $y_i = \mathbf{w}_i^{\mathrm{T}} \mathbf{z}_i$, which one could extend to multivariable output. This module is described in detail in Section 3.3.

3.2. Log-contrasts and sub-compositional coherence

The parts of a composition are fundamentally intertwined. It is impossible to make inferences about the absolute abundance of one variable x_j^* based on the component x_j. This is because the magnitude of x_j depends on all $x_{k \neq j}$. In order to make inferences from relative data that are also true for the absolute data (and thus true in an absolute sense) one *must* use a reference.

By using a ratio of components, the denominator serves as a reference. The ratio a/b describes the abundance of a relative to b (or a conditional upon b (Aitchison and Bacon-Shone, 1984)), while the logarithm is used to center the ratio so that $\log(a/b) = 0$ when $a = b$. Note here that this log-contrast does not depend on the value of c (or c^*). Log-ratios and log-contrasts are *sub-compositionally coherent*: the results do not change when analyzing a subset of the composition (Boogaart and Tolosana-Delgado, 2013b). Any inference made on a ratio of relative data will agree with an inference made on a ratio of absolute data.

A l*og-contrast model* (Aitchison and Bacon-Shone, 1984) is simply a linear model of the log-transformed data:

$$y_i = \beta_0 + \sum_{d=1}^{D} \beta_d \log x_{id} \tag{5}$$

By forcing that $\sum_{d=1}^{D} \beta_d = 0$, the log-contrast becomes more interpretable because the coefficients of the numerator and denominator both have equal weight.

Our network contains a *log-bottleneck module* capable of learning useful log-contrasts. This module passes a log-transform of the input data through a bottleneck layer of a single hidden unit.

$$z_{ib} = \beta_0^{(b)} + \beta_1^{(b)} \log x_{i1} + ... + \beta_D^{(b)} \log x_{iD} \tag{6}$$

$$= \beta_0^{(b)} + \log \left(\prod_{d=1}^{D} (x_{id})^{\beta_d^{(b)}} \right) \tag{7}$$

for sample i and bottleneck b, where z_{ib} is a single hidden unit. By adding a loss that constrains $\sum_{d=1}^{D} \beta_d^{(b)} = 0$, then z_{ib} approximates a log-contrast where the neural network weights refer to the powers (i.e., because $a \log(b) = \log(b^a)$). One could repeat $b = 1...B$ bottlenecks in parallel, depending on the number of log-contrasts required. A regularization penalty determines how many variables comprise each log-contrast, where a larger penalty results in fewer parts. Note, each log-bottleneck resembles the *coda-lasso* model from (Susin et al., 2020).

3.3. Linearity and self-explanation

One advantage of a neural network over a simple linear model is that it can learn a complex non-linear mapping between multivariable input and multivariable output. However, linear models are often preferred in the applied setting because the weights can be interpreted directly as a kind of feature importance. A *(generalized) linear model* (GLM) (Nelder and Wedderburn, 1972) has the form

$$y_i = \Phi\big(w_0 + w_1 \mathbf{x}_{i1} + ... + w_D \mathbf{x}_{iD}\big) \tag{8}$$

where $\mathbf{w}_d$ is the weight for feature d and Φ is a monotonic transform. For example, Φ may be a logistic function when y is binary. By transforming the entire output of the linear model, the “link” function Φ expands the solution space, thus “generalizing” the model. A *generalized additive model* (GAM) (Hastie and Tibshirani, 1986) expands the solution space further, allowing for a more complex mapping between the input and output. A GAM has the form

$$y_i = \Phi\big(w_0 + f_1(\mathbf{x}_{i1}) + ... + f_D(\mathbf{x}_{iD})\big) \tag{9}$$

where f_d is a non-linear function of a single input variable x_{id}. A GAM is potentially more powerful than a GLM, yet still interpretable because each weight describes the importance of a single feature (or a function thereof). However, this model can only solve equations in which the predictor variables make an independent contribution to the output. As written, a GAM cannot model the multiplicative interactions between two or more input variables, e.g., where $y = x_1 + x_2 + x_1 * x_2$.

On the other hand, a fully-connected neural network can learn the interactions between variables without having to specify them explicitly. However, neural networks are hard to interpret because they involve a series of non-linearly transformed matrix products, making it difficult to know how or why a feature contributed to the prediction.

To overcome the disadvantages of GLM, GAM, and neural networks, we introduce a *self-explanation module* which adapts the key idea behind the self-explaining neural network (SENN) (Alvarez-Melis and Jaakkola, 2018). Our network uses self-explanation to introduce non-linearity into the model while also providing personalized interpretability. It has the form

$$y_i = \theta(\mathbf{x}_i)^{\mathrm{T}} \cdot \mathbf{x}_i \tag{10}$$

$$= w_{i1} x_{i1} + ... + w_{iD} x_{iD} \tag{11}$$

for sample i, where θ is a neural network. Here, x_{id} is the abundance of feature d for sample i while w_{id} is the weight of feature d for sample i. Note that for GLM and GAM, each weight w_d is defined for an entire population. For self-explanation, each weight w_{id} is defined for an individual sample. As such, we can interpret w_{id} as a kind of personalized importance score.

We apply self-explanation to the log-contrasts z_{ib}:

$$y_i = \theta(\mathbf{z}_i)^{\mathrm{T}} \cdot \mathbf{z}_i \tag{12}$$

$$= w_{i1} z_{i1} + ... + w_{iB} z_{iB} \tag{13}$$

Alvarez-Melis and Jaakkola (2018) proposed a regularization penalty that forces the self-explanation to act locally linear. We did not include this penalty, and instead throttled the complexity of θ. This design choice reduces the number of parameters and hyper-parameters, making it easier to learn with so few samples.

3.4. Loss

The network is fully differentiable and trained end-to-end. The loss has 3 parts: the mean-squared error, the log-contrast constraint, and the L1-norm regularization penalty.

$$\mathcal{L}_T = \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 + \sum_{b=1}^{B} \lambda_c (\sum_{d=1}^{D} \beta_{db})^2 + \sum_{b=1}^{B} \lambda_s (\sum_{d=1}^{D} |\beta_{db}|) \tag{14}$$

where the weights β_{db} come from the log-bottleneck f_z. Here, λ_c is very large to force the weights of the log-contrast to sum to 0, while λ_s makes it so that each log-contrast contains fewer parts. The number of log-bottlenecks B and the complexity of each log-contrast λ_s are two important hyper-parameters for this model. We set $\lambda_c = 1$.

4. Experiments

4.1. Data and baselines

4.1.1. Synthetic data

We simulate 2 synthetic data sets, both containing 1000 samples belonging to 2 classes ("case" vs. "control"). For the simulated data, we generate the absolute abundances, then divide each sample by its total sum to obtain the relative abundances. For real data, we do not know the absolute abundances, so the simulated data grants us a unique opportunity to compare absolute and relative data analyses.

The first simulated data set is a toy example. It contains 4 variables that represent different bacteria within the gut of a patient, such that the over-proliferation of bacteria $\{2, 3, 4\}$ cause a disease. Our task is to predict whether the simulated patient has a sick or healthy gut.

The second is based on a real biological example in which a mutation of the c-Myc gene causes a cell to massively increase the production of 90% of its gene transcripts, while keeping 10% the same (Lovén et al., 2012). Our task is to predict whether the simulated cell is c-Myc positive or c-Myc negative. Although the original data contain 1000s of features, our simulated version has only 10 features to make visualization easier.

4.1.2. Real data

We use 25 high-throughput sequencing data sets from several sources to benchmark the **DeepCoDA** framework. These data come from two collections. The first contains 13 data sets from (Quinn and Erb, 2020), curated to benchmark compositional data analysis methods for microbiome and similar data[1]. The second contains 12 data sets from (Vangay et al., 2019), curated to benchmark machine learning methods for microbiome data[2]. These 12 were selected from a larger collection because they each had ≥ 100 samples, ≥ 50 samples in the smallest class, and a binary outcome. Since log-contrasts are undefined if any feature equals zero, we first replace zeros with a very small number (using a method that preserves log-ratio abundance for all ratios without zeros (Palarea-Albaladejo and Martín-Fernández, 2015)). The data sets have a median of 220 samples (IQR: 164-404) and 885 features (IQR: 188-1302). A thorough description of the data is available as a **Supplemental Table**.

We develop the model in two stages. First, we use a "discovery set" of 13 data sets to design the architecture and choose its hyper-parameters. Second, we use a "verification set" of 12 unseen data sets to benchmark the final model. Most real data sets lack sufficient samples to tune hyper-parameters. Hence, we use the two-step discovery/verification procedure to identify a set of hyper-parameters that we can recommend for real-world applications.

We benchmark all models with and without self-explanation, and report the AUC distribution across 20 random 90%-10% training-test set splits.

4.1.3. Baselines

For the simulated data, we train a regularized logistic regression model separately for the absolute and relative abundances (i.e., LASSO, with λ chosen by cross-validation). We compare the absolute value of the coefficients for these models to provide an intuition for why a routine analysis of compositional data can yield spurious interpretations.

For the real data, we benchmark our model against several baseline log-ratio transformations, using either regularized logistic regression or random forest. The transformations include: (a) no transformation, (b) the centered log-ratio (Aitchison, 1986), (c) principal balances (Pawlowsky-Glahn et al., 2011), and (d) distal balances (Quinn and Erb, 2020).

[1] Available from https://zenodo.org/record/3378099/

[2] Available from https://knights-lab.github.io/MLRepo/

4.2. Study 1: Feature attribution is unreliable for compositional data

The simulated data contain absolute abundances and relative abundances. For real data, only the relative abundances are known. Our aim is to develop a model in which the interpretation of the relative data agrees with the absolute data. It is easy to show that this is not the case for even simple models like logistic regression.

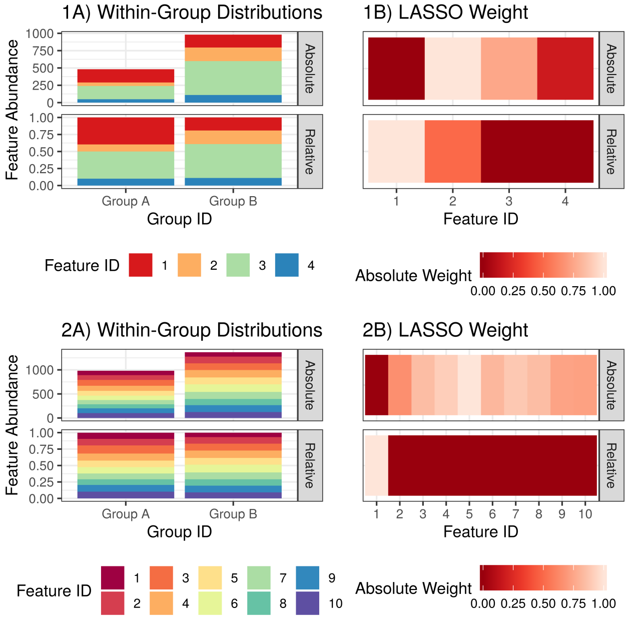

Figure 2. The top and bottom panels correspond to the two simulated data sets. The left panels show a barplot of the absolute and relative abundances. Importantly, both data sets have one feature that does not change between samples (i.e., Feature ID 1), though its proportion does. The right panels show a heatmap of the regression coefficient magnitudes for each feature (min-max scaled, where 1 indicates a large absolute value). Here, we see how the LASSO weights for the relative data disagree completely with those for the absolute data. On the other hand, log-contrasts must be the same for both the absolute and relative data, and so the model coefficients must be the same too.

Figure 2 shows a barplot of the absolute and relative abundances, alongside a heatmap of the regression coefficient magnitudes for each feature. Importantly, both data sets have one feature that does not change between samples (i.e., Feature ID 1), though its proportion does. When fitting a regression on the absolute data, this feature receives zero weight (as it should). However, when fitting a regression on the relative data, this feature receives a large weight. This is because the different groups have a different total abundance. Although this feature has no signal, its relative abundance can perfectly differentiate between the two groups because its relative abundance is conditional upon all other features. Consequently, the coefficient weights swap completely: the zeros become large and the large become zero. The coefficients will change further when working with a subset of the composition; this is especially a problem for microbiome data because the sequencing assay may not measure all microorganisms (and thus one is almost always analyzing a sub-composition).

On the other hand, log-contrasts must be the same for both the absolute and relative data, and so the model coefficients must be the same too. For example, consider the absolute measurements $[a = 4, b = 10, c = 6]$, having the closed proportions $[a = 0.2, b = 0.5, c = 0.3]$. In both cases, the log-contrasts (e.g., $\log(\sqrt{ab}/c)$) are identical.

4.3. Study 2: DeepCoDA achieves state-of-the-art performance but adds personalization

The data sets currently available for microbiome research typically contain only 100s of samples, making hyper-parameter tuning infeasible. For this reason, we sought to identify a set of hyper-parameters that achieved good performance on a collection of data sets. Using a "discovery set" of 13 data sets, we trained models with $B = [1, 3, 5, 10]$ log-bottlenecks and a $\lambda_s = [0.001, 0.01, 0.1, 1]$ L1 penalty. Figure 3 shows the standardized performance for all "discovery set" models for each hyper-parameter combination. Here, we see that $B = 5$ and $\lambda_s = 0.01$ works well with or without self-explanation. **Supplemental Figure 1** shows the **DeepCoDA** performance compared with the baselines, where our model achieves appreciable performance.

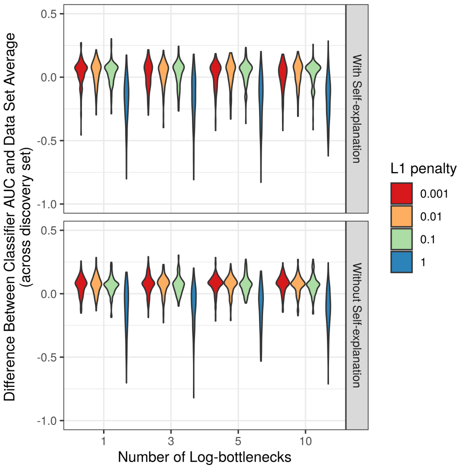

Figure 3. This figure shows the relative performance of our model for each combination of hyper-parameters, with and without self-explanation (using the "discovery set" only). The y-axis shows the distribution of performances, standardized so that the performances from one data set are centered around the data set average.

Next, we used a "verification set" of 12 unseen data sets to ensure that the selected hyper-parameters generalize to new data. Figure 4 shows the **DeepCoDA** performance

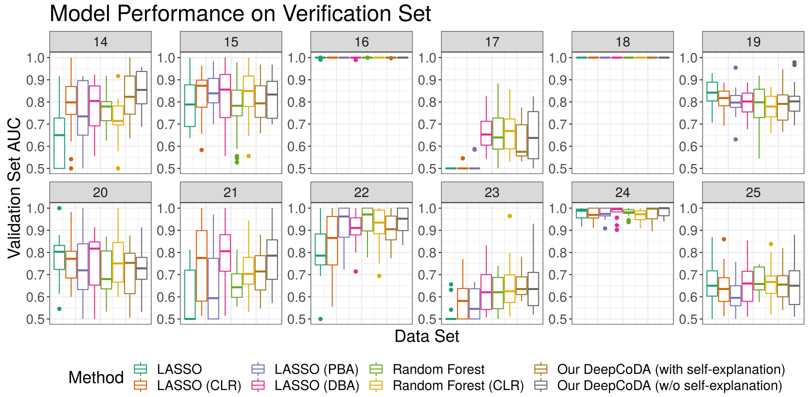

Figure 4. This figure shows the AUC for several models, organized by the method (x-axis) and data set source (facet). For all models, the boxplot shows the AUC distribution across 20 random 90%-10% training-test set splits. All **DeepCoDA** models use 5 log-bottlenecks and an L1 penalty of 0.01, chosen based on the "discovery set". Our model achieves appreciable performance across the 25 data sets. See the **Supplement** for all results from data sets 1-13.

compared with the baselines. Again, we see that our model achieves appreciable performance. With only 100s of samples, it is not surprising that the neural network does not clearly outperform linear models; however, our aim is not to improve performance, but to extend personalized interpretability to compositional data.

Although most biomarker data sets only contain 100s of samples, **Supplemental Figure 2** shows that **DeepCoDA** can scale to larger data sets with 1000s of samples.

4.4. Study 3: DeepCoDA produces transparent models

To better understand how our model works for real data, we analyze the network weights and layer activations for data set "3" (comparing inflammatory bowel disease with healthy controls (Franzosa et al., 2019)). Having already established the model's performance, we analyze the model as trained on the entire data set of 220 samples.

Figure 5 illustrates how the model makes a prediction. The top row has 5 panels, one for each log-bottleneck, that plot the patient-specific weight for each log-contrast w_{ib} (y-axis) versus the value of the log-contrast itself z_{ib} (x-axis). The bottom row also has 5 panels, showing the training set ROC for the product $w_{ib} * z_{ib}$. For **DeepCoDA** with self-explanation, the final prediction is $y_i = \Phi(\sum_{b=1}^{5} w_{ib} * z_{ib})$. This formulation allows us to describe in simple algebraic terms how the classifier made each prediction, bringing transparency to the decision-making process.

Although the principal advantage of our model is personalized interpretability, we can analyze the patient-specific weights to better understand how the model "sees" the data. When comparing the patient-specific weights w_b with the log-contrasts z_b, there are (at least) 3 possibilities: (a) w_b is uniform, meaning that the importance of a log-contrast is the same for all samples, (b) w_b depends on z_b, meaning that the importance of a log-contrast depends on itself (i.e., a non-linear transform of z_b), (c) w_b depends on other log-contrasts (i.e., a statistical interaction with z_b).

In Figure 5, we see that log-contrast 4 is an example of (b). As the "balance" between the numerator and denominator "tips" toward the numerator, the log-contrast receives a higher positive weight. Thus, the product $w_4 * z_4 \approx (z_4)^2$. On the other hand, log-contrast 5 is an example of (c). This is shown by the canonical correlation between the patient-specific weights and the log-contrast values, where we see that w_5 is strongly correlated with z_3, implying that the importance of z_5 depends on z_3. Thus, the product $w_5 * z_5 \approx z_3 * z_5$ (because $w_5 \approx z_3$), suggesting a log-multiplicative interaction within the log-additive model.

Although each outcome is predicted by a linear combination of the input, the distribution of sample weights can be correlated with the sample input to reveal higher-order biological interactions. Therefore, our model not only realizes personalized interpretability through patient-specific weights, but can be studied post-hoc to reveal *how* the model generated the weights.

Our results suggest that self-explanation can produce highly interpretable linear estimates from familiar polynomial expressions, such as power-law transformations and statistical interactions. This finding makes a clear connection between self-explanation and classical statistics, and may help elicit trust from clinicians who are skeptical of deep learning.

4.5. Study 4: 2-levels of interpretability

The **DeepCoDA** framework offers 2-levels of interpretability: (1) the "weights" of the self-explanation module tell us how the classifier predicts a class label; (2) the weights of

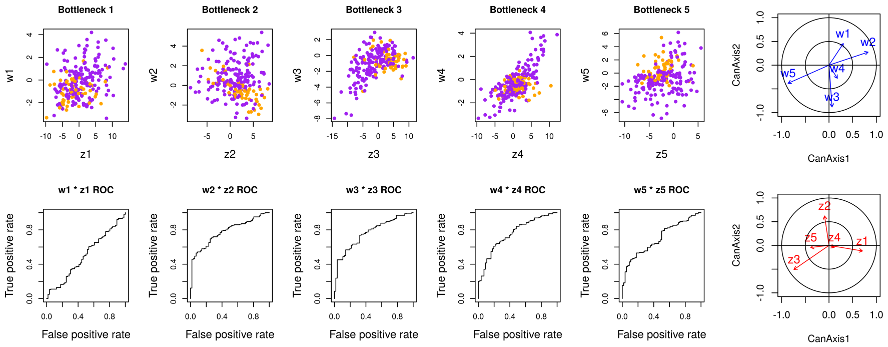

Figure 5. This figure illustrates how the model makes a prediction (using data set "3" as a case study). The top row has 5 panels, one for each log-bottleneck, that plot the patient-specific weight for each log-contrast w_{ib} (y-axis) versus the value of the log-contrast itself z_{ib} (x-axis). The bottom row also has 5 panels, showing the training set ROC for the product $w_{ib} * z_{ib}$. The right-most column shows a canonical correlation analysis between the patient-specific weights and log-contrast values.

the log-bottleneck module tell us which features contribute to each log-contrast.

At the first level, the product scores ($w_i * z_i$) can be interpreted directly by a clinical laboratory or researcher to identify which features drive the final prediction. When a classifier makes a decision, the patient-specific weights (w_i) are multiplied with the log-contrast values (z_i), then added together. In data set "3", a patient is predicted to have inflammatory bowel disease if this sum exceeds zero; otherwise, the patient is healthy. The largest product scores contribute most to the decision.

At the second level, the log-bottleneck *weights* define how each bacteria contribute to the log-contrasts. For log-contrast 3, the bacteria *Gordonibacter pamelaeae* makes the largest contribution to the numerator, while *Bacteroides cellulosilyticus* makes the largest contribution to the denominator. Meanwhile, the signs of the log-contrasts reveal which bacteria dominate. Negative values mean that the denominator bacteria outweigh those in the numerator; positives mean that the numerator outweighs the denominator.

We expand this discussion in the **Supplement**.

5. Conclusion

The **DeepCoDA** framework achieves personalized interpretability through patient-specific weights, while using log-contrasts to ensure that any interpretation is coherent for compositional data. Our model had appreciable performance across 25 data sets when compared with well-known baselines, and can even perform well without any hyper-parameter tuning. However, our aim is not to improve performance, but to extend personalized interpretability to compositional data. For **DeepCoDA**, self-explanation allows us to describe in simple algebraic terms how the classifier made each prediction, bringing transparency to the decision-making process.

We introduced the log-bottleneck as a new neural network architecture to learn log-contrasts through gradient descent. However, even with L1 regularization, the coefficients for the learned log-contrasts were not very sparse (e.g., some log-contrasts for data set "3" contained 144/153 parts). Fortunately, the distribution of powers is not uniform within a log-contrast, making it possible to summarize a log-contrast by its most prominent members. Still, simpler log-contrasts are preferable in most cases. Future work should examine how best to innovate the log-bottleneck to achieve simpler log-contrasts, for example through L0 regularization or a discrete parameter search.

6. Availability of data and materials

Our implementation of **DeepCoDA** is available from `http://github.com/nphdang/DeepCoDA`. The scripts used to synthesize data, apply baselines, and visualize results are available from `http://doi.org/10.5281/zenodo.3893986`.

7. Acknowledgements

This research was partially funded by the Australian Government through the Australian Research Council (ARC). Prof Venkatesh is the recipient of an ARC Australian Laureate Fellowship (FL170100006).

References

J Aitchison. *The Statistical Analysis of Compositional Data*. Chapman & Hall, Ltd., London, UK, UK, 1986. ISBN 978-0-412-28060-3.

J. Aitchison and J. Bacon-Shone. Log contrast models for experiments with mixtures. *Biometrika*, 71(2):323–330, August 1984. ISSN 0006-3444. doi: 10.1093/biomet/71.2.323.

J. Aitchison, C. Barceló-Vidal, J. A. Martín-Fernández, and V. Pawlowsky-Glahn. Logratio Analysis and Compositional Distance. *Mathematical Geology*, 32(3):271–275, April 2000. ISSN 0882-8121, 1573-8868. doi: 10.1023/A:1007529726302.

David Alvarez-Melis and Tommi S. Jaakkola. Towards Robust Interpretability with Self-Explaining Neural Networks. *arXiv:1806.07538 [cs, stat]*, December 2018. arXiv: 1806.07538.

Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11: R106, October 2010. ISSN 1474-760X. doi: 10.1186/gb-2010-11-10-r106.

Euan A. Ashley. Towards precision medicine. *Nature Reviews Genetics*, 17(9):507–522, September 2016. ISSN 1471-0064. doi: 10.1038/nrg.2016.86.

Marta Avalos, Richard Nock, Cheng Soon Ong, Julien Rouar, and Ke Sun. Representation Learning of Compositional Data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6679–6689. Curran Associates, Inc., 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, May 2016. arXiv: 1409.0473.

Adham Beykikhoshk, Thomas P. Quinn, Samuel C. Lee, Truyen Tran, and Svetha Venkatesh. DeepTRIAGE: interpretable and individualised biomarker scores using attention mechanism for the classification of breast cancer subtypes. *BMC Medical Genomics*, 13(3):20, February 2020. ISSN 1755-8794. doi: 10.1186/s12920-020-0658-5.

K. Gerald van den Boogaart and Raimon Tolosana-Delgado. Descriptive Analysis of Compositional Data. In *Analyzing Compositional Data with R*, Use R!, pages 73–93. Springer, Berlin, Heidelberg, 2013a. ISBN 978-3-642-36808-0 978-3-642-36809-7. doi: 10.1007/978-3-642-36809-7_4.

K. Gerald van den Boogaart and Raimon Tolosana-Delgado. Fundamental Concepts of Compositional Data Analysis. In *Analyzing Compositional Data with R*, Use R!, pages 13–50. Springer Berlin Heidelberg, 2013b. ISBN 978-3-642-36808-0 978-3-642-36809-7. doi: 10.1007/978-3-642-36809-7_2.

M. Luz Calle. Statistical Analysis of Metagenomics Data. *Genomics & Informatics*, 17(1), March 2019. ISSN 2234-0742. doi: 10.5808/GI.2019.17.1.e6.

Claire Duvallet, Sean M. Gibbons, Thomas Gurry, Rafael A. Irizarry, and Eric J. Alm. Meta-analysis of gut microbiome studies identifies disease-specific and shared responses. *Nature Communications*, 8(1):1784, December 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-01973-8.

Ionas Erb and Cedric Notredame. How should we measure proportionality on relative gene expression data? *Theory in Biosciences*, 135:21–36, 2016. ISSN 1431-7613. doi: 10.1007/s12064-015-0220-8.

Eric A. Franzosa, Alexandra Sirota-Madi, Julian Avila-Pacheco, Nadine Fornelos, Henry J. Haiser, Stefan Reinker, Tommi Vatanen, A. Brantley Hall, Himel Mallick, Lauren J. McIver, Jenny S. Sauk, Robin G. Wilson, Betsy W. Stevens, Justin M. Scott, Kerry Pierce, Amy A. Deik, Kevin Bullock, Floris Imhann, Jeffrey A. Porter, Alexandra Zhernakova, Jingyuan Fu, Rinse K. Weersma, Cisca Wijmenga, Clary B. Clish, Hera Vlamakis, Curtis Huttenhower, and Ramnik J. Xavier. Gut microbiome structure and metabolic activity in inflammatory bowel disease. *Nature Microbiology*, 4(2):293–305, 2019. ISSN 2058-5276. doi: 10.1038/s41564-018-0306-4.

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

Trevor Hastie and Robert Tibshirani. Generalized Additive Models. *Statistical Science*, 1(3):297–310, August 1986. ISSN 0883-4237, 2168-8745. doi: 10.1214/ss/1177013604.

Vuong Le, Thomas P. Quinn, Truyen Tran, and Svetha Venkatesh. Deep in the Bowel: Highly Interpretable Neural Encoder-Decoder Networks Predict Gut Metabolites from Gut Microbiome. *bioRxiv*, page 686394, June 2019. doi: 10.1101/686394.

David Lovell, Vera Pawlowsky-Glahn, Juan José Egozcue, Samuel Marguerat, and Jürg Bähler. Proportionality: A Valid Alternative to Correlation for Relative Data. *PLoS Computational Biology*, 11(3), March 2015. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1004075.

Jakob Lovén, David A. Orlando, Alla A. Sigova, Charles Y. Lin, Peter B. Rahl, Christopher B. Burge, David L. Levens, Tong Ihn Lee, and Richard A. Young. Revisiting Global Gene Expression Analysis. *Cell*, 151

(3):476–482, October 2012. ISSN 0092-8674. doi: 10.1016/j.cell.2012.10.012.

Josep A. Martín-Fernández, Mark A. Engle, Leslie F. Ruppert, and Ricardo A. Olea. Advances in self-organizing maps for their application to compositional data. *Stochastic Environmental Research and Risk Assessment*, February 2019. ISSN 1436-3259. doi: 10.1007/s00477-019-01659-1.

James T. Morton, Jon Sanders, Robert A. Quinn, Daniel McDonald, Antonio Gonzalez, Yoshiki Vázquez-Baeza, Jose A. Navas-Molina, Se Jin Song, Jessica L. Metcalf, Embriette R. Hyde, Manuel Lladser, Pieter C. Dorrestein, and Rob Knight. Balance Trees Reveal Microbial Niche Differentiation. *mSystems*, 2(1):e00162–16, February 2017. ISSN 2379-5077. doi: 10.1128/mSystems.00162-16.

James T. Morton, Alexander A. Aksenov, Louis Felix Nothias, James R. Foulds, Robert A. Quinn, Michelle H. Badri, Tami L. Swenson, Marc W. Van Goethem, Trent R. Northen, Yoshiki Vazquez-Baeza, Mingxun Wang, Nicholas A. Bokulich, Aaron Watters, Se Jin Song, Richard Bonneau, Pieter C. Dorrestein, and Rob Knight. Learning representations of microbe–metabolite interactions. *Nature Methods*, pages 1–9, November 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0616-3.

J. A. Nelder and R. W. M. Wedderburn. Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370, 1972. ISSN 00359238. doi: 10.2307/2344614.

Javier Palarea-Albaladejo and Josep Antoni Martín-Fernández. zCompositions — R package for multivariate imputation of left-censored data under a compositional approach. *Chemometrics and Intelligent Laboratory Systems*, 143:85–96, April 2015. ISSN 0169-7439. doi: 10.1016/j.chemolab.2015.02.019.

Vera Pawlowsky-Glahn, Juan José Egozcue, and Raimon Tolosana Delgado. Principal balances. *Proceedings of CoDaWork 2011, The 4th Compositional Data Analysis Workshop*, pages 1–10, 2011.

Karl Pearson. Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 187:253–318, 1896. ISSN 0264-3952.

Thomas P. Quinn and Ionas Erb. Interpretable Log Contrasts for the Classification of Health Biomarkers: a New Approach to Balance Selection. *mSystems*, 5(2), April 2020. ISSN 2379-5077. doi: 10.1128/mSystems.00230-19.

Thomas P. Quinn, Ionas Erb, Mark F. Richardson, and Tamsyn M. Crowley. Understanding sequencing data as compositions: an outlook and review. *Bioinformatics*, 34 (16):2870–2878, August 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty175.

J. Rivera-Pinto, J. J. Egozcue, V. Pawlowsky-Glahn, R. Paredes, M. Noguera-Julian, and M. L. Calle. Balances: a New Perspective for Microbiome Analysis. *mSystems*, 3(4):e00053–18, August 2018. ISSN 2379-5077. doi: 10.1128/mSystems.00053-18.

Mark D. Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11:R25, 2010. ISSN 1474-760X. doi: 10.1186/gb-2010-11-3-r25.

W. B. Schwartz, R. S. Patil, and P. Szolovits. Artificial intelligence in medicine. Where do we stand? *The New England Journal of Medicine*, 316(11):685–688, March 1987. ISSN 0028-4793. doi: 10.1056/NEJM198703123161109.

Justin D. Silverman, Alex D. Washburne, Sayan Mukherjee, and Lawrence A. David. A phylogenetic transform enhances analysis of compositional microbiota data. *eLife*, 6, 2017. ISSN 2050-084X. doi: 10.7554/eLife.21887.

Antoni Susin, Yiwen Wang, Kim-Anh Lê Cao, and M. Luz Calle. Variable selection in microbiome compositional data analysis. *NAR Genomics and Bioinformatics*, 2(2), June 2020. doi: 10.1093/nargab/lqaa029.

Raimon Tolosana Delgado, Hassan Talebi, Mahdi Khodadadzadeh, and K. Gerald van den Boogaart. On machine learning algorithms and compositional data. *CoDaWork2019*, 2019.

Michael Tsang, Hanpeng Liu, Sanjay Purushotham, Pavankumar Murali, and Yan Liu. Neural Interaction Transparency (NIT): Disentangling Learned Interactions for Improved Interpretability. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5804–5813. Curran Associates, Inc., 2018.

Pajau Vangay, Benjamin M. Hillmann, and Dan Knights. Microbiome Learning Repo (ML Repo): A public repository of microbiome regression and classification tasks. *GigaScience*, 8(5), May 2019. doi: 10.1093/gigascience/giz042.

Alex D. Washburne, Justin D. Silverman, Jonathan W. Leff, Dominic J. Bennett, John L. Darcy, Sayan Mukherjee, Noah Fierer, and Lawrence A. David. Phylogenetic factorization of compositional data yields lineage-level associations in microbiome datasets. *PeerJ*, 5:e2969, 2017. ISSN 2167-8359. doi: 10.7717/peerj.2969.