Learning Sparse Log-Ratios for High-Throughput Sequencing Data

Elliott Gordon-Rodriguez 1  Thomas P. Quinn 2  John P. Cunningham 1

Abstract

The automatic discovery of interpretable features that are associated to an outcome of interest is a central goal of bioinformatics. In the context of high-throughput genetic sequencing data, and Compositional Data more generally, an important class of features are the log-ratios between subsets of the input variables. However, the space of these log-ratios grows combinatorially with the dimension of the input, and as a result, existing learning algorithms do not scale to increasingly common high-dimensional datasets. Building on recent literature on continuous relaxations of discrete latent variables, we design a novel learning algorithm that identifies sparse log-ratios several orders of magnitude faster than competing methods. As well as dramatically reducing runtime, our method outperforms its competitors in terms of sparsity and predictive accuracy, as measured across a wide range of benchmark datasets.

1. Introduction

Much recent work has been devoted to designing differentiable relaxations of discrete latent variables. These relaxations can be used to learn class membership (Jang et al., 2016; Maddison et al., 2017; Potapczynski et al., 2020), permutations (Linderman et al., 2018; Mena et al., 2018), subsets (Xie & Ermon, 2019; Yang et al., 2019), and rankings (Cuturi et al., 2019; Blondel et al., 2020). Depending on their use case, existing methods range in complexity, from the simple-but-effective straight-through estimator (Bengio et al., 2013), to mathematically intricate schemes based on optimal transport (Xie et al., 2020). However, the driving principle is always the same: to enable efficient gradient-based optimization on an otherwise intractable discrete space. The goal of our work is to extend this principle to a novel setting where, to the best of our knowledge, no differentiable relaxations have yet been proposed. Motivated by domain applications, our objective is to select log-ratios from a set of covariates, a problem that is equivalent to a discrete optimization over pairs of disjoint subsets (where the pair represents the numerator and denominator of the ratio, respectively). Our novel relaxation will result in dramatic speedups over several recent state-of-the-art learning algorithms from the field of bioinformatics, thereby enabling the analysis of much larger datasets than previously possible.

Log-ratios are an important class of features for analyzing high-throughput sequencing (HTS) metagenomic data (Wooley et al., 2010; Gloor & Reid, 2016; Gloor et al., 2017; Quinn et al., 2018). For example, in microbiome count data, the relative weight between two sub-populations of related microorganisms can serve as a clinically useful biomarker (Rahat-Rozenbloom et al., 2014; Crovesy et al., 2020; Magne et al., 2020). More generally, log-ratios are of fundamental importance to the field of Compositional Data (CoDa), of which HTS data can be seen as a special case (Pawlowsky-Glahn & Egozcue, 2006; Pawlowsky-Glahn & Buccianti, 2011). CoDa can be defined as simplex-valued data, or equivalently, non-negative vectors whose totals are uninformative, i.e., relative data. Due to the nature of the recording technique, HTS data represents the relative abundance of different microbial signatures in a given sample, and therefore is an instance of CoDa (Gloor & Reid, 2016; Gloor et al., 2017; Quinn et al., 2018). Indeed, the application of CoDa methodology to HTS data has become increasingly popular in recent years (Fernandes et al., 2013; 2014; Rivera-Pinto et al., 2018; Quinn et al., 2019; Calle, 2019), with log-ratios serving as the basic building blocks for statistical analysis.

But why do log-ratios form the basis of CoDa methodology? Unlike unconstrained real-valued data, the relative nature of HTS data and CoDa results in each covariate becoming negatively correlated to all others (increasing one component of a composition implies a relative decrease of the other components). It is well known that, as a result, the usual measures of association and feature attribution are problematic when applied to CoDa (Pearson, 1896; Filzmoser et al., 2009; Van den Boogaart & Tolosana-Delgado, 2013; Lovell et al., 2015). Log-ratios account for this idiosyncratic structure by transforming CoDa onto unconstrained feature space, where the usual tools of statistical learning apply (Aitchison, 1982; Pawlowsky-Glahn & Egozcue, 2006). The choice

1Department of Statistics, Columbia University 2Applied Artificial Intelligence Institute (A2I2), Deakin University. Correspondence to: Elliott Gordon-Rodriguez <eg2912@columbia.edu>.
Learning Sparse Log-Ratios for High-Throughput Sequencing Data

of the log-ratio transform offers the necessary property of scale invariance, but in the CoDa literature it holds primacy for a variety of other technical reasons, including so-called subcompositional coherence (Aitchison, 1982; Pawlowsky-Glahn & Buccianti, 2011; Egozcue & Pawlowsky-Glahn, 2019). Log-ratios can be taken over pairs of individual covariates (Aitchison, 1982; Greenacre, 2019b) or aggregations thereof, typically geometric means (Aitchison, 1982; Egozcue et al., 2003; Egozcue & Pawlowsky-Glahn, 2005; Rivera-Pinto et al., 2018; Quinn & Erb, 2019) or summations (Greenacre, 2019a; 2020; Quinn & Erb, 2020). The resulting features work well empirically, but also imply a clear interpretation: a log-ratio is a single composite score that expresses the overall quantity of one sub-population as compared with another. When the log-ratios are sparse, meaning they are taken over a small number of covariates, they define biomarkers that are particularly intuitive to understand, a key desiderata for predictive models that are of clinical relevance (Goodman & Flaxman, 2017).

Thus, learning sparse log-ratios is a central problem in CoDa. This problem is especially challenging in the context of HTS data, due to its high dimensionality (ranging from 100 to over 10,000 covariates). Existing methods rely on stepwise search or evolutionary algorithms (Rivera-Pinto et al., 2018; Greenacre, 2019b; Quinn & Erb, 2020; Pritfi et al., 2020), which scale very poorly with the dimension of the input. These algorithms are prohibitively slow for most HTS datasets, and thus there is a new demand for sparse and interpretable models that scale to high dimensions (Li, 2015; Cammarota et al., 2020; Susin et al., 2020).

This demand motivates the present work, in which we present CoDaCoRe,1 a novel learning algorithm for Compositional Data via Continuous Relaxations. The key idea behind CoDaCoRe is to approximate a combinatorial optimization over the set of log-ratios (equivalent to the set of pairs of disjoint subsets of the covariates), by means of a continuous relaxation that can be optimized efficiently using gradient descent. To the best of our knowledge, CoDaCoRe is the first CoDa method that scales to high dimensions, and that simultaneously produces sparse, interpretable, and accurate models. The main contributions of our method can be summarized as follows:

- **Computational efficiency.** CoDaCoRe scales linearly with the dimension of the input. It runs several orders of magnitude faster than its competitors.
- **Interpretability.** CoDaCoRe identifies a set of log-ratios that are sparse, biologically meaningful, and ranked in order of importance. Our model is highly interpretable, and much sparser, relative to competing methods of similar accuracy and computational complexity.
- **Predictive accuracy.** CoDaCoRe achieves better out-of-sample accuracy than existing CoDa methods, and performs similarly to state-of-the-art black-box classifiers (which are neither sparse nor interpretable).
- **Optimization robustness.** We leverage the functional form of our continuous relaxation to identify an adaptive learning rate that enables CoDaCoRe to converge reliably, requiring no additional hyperparameter tuning when deployed on novel datasets.

2. Background

Our work focuses on the supervised learning problem with compositional predictors. Namely, we are given data \( \{x_i, y_i\}_{i=1}^n \), where \( x_i \) is compositional (e.g., HTS data), and our goal is to learn an association \( x_i \mapsto y_i \). For many microbiome applications, \( x_i \) represents a vector of frequencies of the different species of bacteria that compose the microbiome of the \( i \)th subject. In other words, \( x_{ij} \) denotes the abundance of the \( j \)th species (of which there are \( p \) total) in the \( i \)th subject. The response \( y_i \) is a binary variable indicating whether the \( i \)th subject belongs to the case or the control groups (e.g., sick vs. healthy). Due to the nature of HTS, the input frequencies \( x_{ij} \) arise from an inexhaustive sampling procedure, so that the totals \( \sum_{j=1}^p x_{ij} \) are arbitrary and the components should only be interpreted in relative terms (i.e., as CoDa) (Gloor & Reid, 2016; Gloor et al., 2017; Quinn et al., 2018; Calle, 2019). While we mainly consider applications to microbiome data, our method applies more generally to any high-dimensional CoDa, including those produced by Liquid Chromatography Mass Spectrometry (Filzmoser & Walczak, 2014).

In order to account for the compositional nature of \( x_i \), we seek log-ratio transformed features that can be passed to a regression function downstream. As discussed, these log-ratios will result in interpretable features and scale-invariant models (that are also subcompositionally coherent). The simplest such choice is to take the pairwise log-ratios between input variables, i.e., \( \log(x_{ij}^+/x_{ij}^-) \), where \( (j^+, j^-) \) indexes a pair of covariates (Aitchison, 1982). Note that the ratio cancels out any scaling factor applied to \( x_i \), preserving only the relative information in the data, while the log transform ensures the output is (unconstrained) real-valued. In order to select a good pair \( (j^+, j^-) \) from the input covariates, Greenacre (2019b) proposed a step-wise algorithm for identifying pairwise log-ratios that explain the most variation in a dataset. This algorithm produces a sparse and interpretable set of features, but it is prohibitively slow on high-dimensional datasets, as a result of the step-wise search scaling poorly in the dimension of the input. A heuristic search algorithm that is less accurate but computationally faster has been developed as part of Quinn et al. (2017),

---

1Our implementation can be downloaded from https://github.com/cunningham-lab/codacore.
though its computational cost is still troublesome (as we shall see in Section 4).

2.1. Balances

Recently, a class of log-ratios known as balances (Egozcue & Pawlowsky-Glahn, 2005) have become of interest in microbiome applications, due to their interpretability as the relative weight between two sub-populations of bacteria (Morton et al., 2019b; Quinn & Erb, 2019). Balances are defined as the log-ratios between geometric means of two subsets of the covariates:

\[
B(x_i; J^+, J^-) = \log \left( \frac{\prod_{j \in J^+} x_{ij}^{p^+}}{\prod_{j \in J^-} x_{ij}^{p^-}} \right)
\]

\[
= \frac{1}{p^+} \sum_{j \in J^+} \log x_{ij} - \frac{1}{p^-} \sum_{j \in J^-} \log x_{ij},
\]

where \(J^+\) and \(J^-\) denote a pair of disjoint subsets of the indices \(\{1, \ldots, p\}\), and \(p^+\) and \(p^-\) denote their respective sizes. For example, in microbiome data, \(J^+\) and \(J^-\) are groups of bacteria species that may be related by their environmental niche (Morton et al., 2017) or genetic similarity (Silverman et al., 2017; Washburne et al., 2017). Note that when \(p^+ = p^- = 1\) (i.e., \(J^+\) and \(J^-\) each contain a single element), \(B(x_i; J^+, J^-)\) reduces to a pairwise log-ratio. By allowing for the aggregation of more than one covariate in the numerator and denominator of the log-ratio, balances provide a richer set of features that allows for more flexible models than pairwise log-ratios. Insofar as the balances are taken over a small number of covariates (i.e., \(J^+\) and \(J^-\) are sparse), they also provide highly interpretable biomarkers.

The selbal algorithm (Rivera-Pinto et al., 2018) has gained popularity as a method for automatically identifying balances that predict a response variable. However, this algorithm is also based on a step-wise search through the combinatorial space of subset pairs \((J^+, J^-)\), which scales poorly in the dimension of the input and becomes prohibitively slow for HTS data (Susin et al., 2020).

2.2. Amalgamations

An alternative to balances, known as amalgamations, is defined by aggregating components through summation:

\[
A(x_i; J^+, J^-) = \log \left( \frac{\sum_{j \in J^+} x_{ij}}{\sum_{j \in J^-} x_{ij}} \right),
\]

where again \(J^+\) and \(J^-\) denote disjoint subsets of the input components. Amalgamations have the advantage of reduc-

ing the dimensionality of the data through an operation, the sum, that some authors argue is more interpretable than a geometric mean (Greenacre, 2019a; Greenacre et al., 2020). On the other hand, amalgamations can be less effective than balances for identifying components that are statistically important, but small in magnitude, e.g., rare bacteria species (since small terms will have less impact on a summation than on a product).

Recently, Greenacre (2020) has advocated for the use of expert-driven amalgamations, using domain knowledge to construct the relevant features. On the other hand, Quinn & Erb (2020) proposed amalgam, an evolutionary algorithm to automatically identify amalgamated log-ratios (Eq. 2) that are predictive of a response variable. However, this algorithm does not scale to high-dimensional variables (albeit, comparing favorably to selbal), nor does it produce sparse models (hindering interpretability of the results).

2.3. Other Related Work

CoDa methodology has also recently attracted interest from the machine learning community (Tolosana-Delgado et al., 2019; Quinn et al., 2020; Gordon-Rodriguez et al., 2020a;b; Templ, 2020). Relevant to us is DeepCoDA (Quinn et al., 2020), which combines self-explaining neural networks with log-ratio transformed features. In particular, DeepCoDA learns a set of log-contrasts, in which the numerator and denominator are defined as unequally weighted geometric averages of components. As a result of this weighting, DeepCoDA loses much of the interpretability and intuitive appeal of balances (or amalgamations), which is exacerbated by its lack of sparsity (in spite of regularization). Moreover, like most deep architectures, DeepCoDA is sensitive to initialization and optimization hyperparameters (which limits its ease of use) and is susceptible to overfitting (which can further compromise interpretability of the model).

The special case of a linear log-contrast model has been referred to as Coda-lasso, and was separately proposed by Lu et al. (2019). While Coda-lasso scales better than selbal, it has been found to perform worse in terms of predictive accuracy (Susin et al., 2020). More importantly, Coda-lasso is still prohibitively slow on the high-dimensional HTS data that we wish to consider. Last, we highlight another common set of features that are also a special case of log-contrasts: centered-log-ratios, where individual covariates are divided by the geometric mean of all input variables (Aitchison, 1982). Models using these features, such as Susin et al. (2020), can be accurate and computationally efficient, however they are inherently not sparse and are difficult to interpret scientifically (Greenacre, 2019a).
Learning Sparse Log-Ratios for High-Throughput Sequencing Data

Table 1. Qualitative comparison of the methods discussed, ordered from most sparse (top) to least (bottom). CoDaCoRe is the only learning algorithm that performs on all of our criteria. See Table 2 for a corresponding quantitative comparison.

<table>
<thead>
<tr>
<th>Method</th>
<th>Scalability</th>
<th>Interpretability</th>
<th>Sparsity</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CoDaCoRe (Ours)</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Pairwise log-ratios (Greenacre, 2019b)</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Selbal (Rivera-Pinto et al., 2018)</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>LASSO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CODA-LASSO (Lu et al., 2019)</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amalgam (Quinn &amp; Err, 2020)</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DeepCoDA (Quinn et al., 2020)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>CLR-LASSO (Susin et al., 2020)</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Black-box (Random Forest, XGBoost)</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

3. Methods

We now present CoDaCoRe, a novel learning algorithm for HTS data, and more generally, high-dimensional CoDa. Unlike existing methods, CoDaCoRe is simultaneously scalable, interpretable, sparse, and accurate. We compare the relative merits of CoDaCoRe and its competitors in Table 1.

3.1. Continuous Relaxation

In its basic formulation, CoDaCoRe learns a regression function of the form:

$$f(x) = \alpha + \beta \cdot B(x; J^+, J^-),$$  \hspace{1cm} (3)

where $B$ denotes a balance (Eq. 1), and $\alpha$ and $\beta$ are scalar parameters. For clarity, we will restrict our exposition to this formulation, but note that our algorithm can be applied equally to learn amalgamations instead of balances (see Section 3.5), as well as generalizing straightforwardly to nonlinear functions (provided they are suitably parameterized and differentiable).

Let $L(y, f)$ denote the cross-entropy loss, with $f \in \mathbb{R}$ given in logit space. The goal of CoDaCoRe is to find the balance that is maximally associated with the response. Mathematically, this can be written as an empirical risk minimization:

$$\min_{(J^+, J^-, \alpha, \beta)} \sum_i L(y_i, \alpha + \beta \cdot B(x_i; J^+, J^-)).$$  \hspace{1cm} (4)

This objective involves a discrete optimization over pairs $(J^+, J^-)$ of disjoint subsets, a combinatorially hard problem. The key insight of CoDaCoRe is to approximate this combinatorial optimization with a continuous relaxation that can be trained efficiently by gradient descent.

Our relaxation is parameterized by an unconstrained vector of “assignment weights”, $w \in \mathbb{R}^p$, with one scalar parameter per input dimension (e.g., one weight per bacteria species). The weights are mapped to a vector of “soft assignments” via:

$$\tilde{w} = 2 \cdot \text{sigmoid}(w) - 1 = \frac{2}{1 + \exp(-w)} - 1,$$  \hspace{1cm} (5)

where the sigmoid is applied component-wise. Eq. 5 maps onto the interval $(-1, 1)$, which can be understood straightforwardly as a relaxation of the set $\{-1, 1, 0\}$, denoting membership to $J^-$, $J^+$, or neither, respectively. Let us write $\tilde{w}^+ = \text{ReLU}(\tilde{w})$ and $\tilde{w}^- = \text{ReLU}(-\tilde{w})$ for the positive and negative parts of $\tilde{w}$, respectively. We approximate balances (Eq. 1) with the following relaxation:

$$\tilde{B}(x_i; w) = \frac{\sum_j \tilde{w}_j^+ \log x_{ij}}{\sum_j \tilde{w}_j^+} - \frac{\sum_j \tilde{w}_j^- \log x_{ij}}{\sum_j \tilde{w}_j^-},$$  \hspace{1cm} (6)

In other words, we approximate geometric averages over subsets of the inputs, by weighted geometric averages over all components (compare Equations 1 and 6).

Crucially, this relaxation is differentiable in $w$, allowing us to construct a surrogate objective function that can be optimized jointly in $(w, \alpha, \beta)$ by gradient descent:

$$\min_{(w, \alpha, \beta)} \sum_i L(y_i, \alpha + \beta \cdot \tilde{B}(x_i; w)).$$  \hspace{1cm} (8)

We defer the details of our implementation of gradient descent to the Supplement (Section A), but we highlight two observations. First, the computational cost of the gradient of Eq. 8 is linear in the dimension of $w$. As a result, our algorithm scales linearly with the dimension of the input, and is fast to fit on large datasets (see Section 4.3). Second, knowledge of the functional form of our relaxation (Eq. 6) can be exploited in order to select the learning rate adaptively (i.e., without tuning), resulting in robust convergence across all real and simulated datasets that we considered.

3.2. Discretization

While a set of features in the form of Eq. 6 may perform well, it may not be easy to interpret. Crucially, this relaxation is differentiable in $w$, allowing us to construct a surrogate objective function that can be optimized jointly in $(w, \alpha, \beta)$ by gradient descent:

$$\min_{(w, \alpha, \beta)} \sum_i L(y_i, \alpha + \beta \cdot \tilde{B}(x_i; w)).$$  \hspace{1cm} (8)
Learning Sparse Log-Ratios for High-Throughput Sequencing Data

all covariates is much harder for a biologist to interpret (and less intuitively appealing) than a bona fide balance over a small number of covariates. On these grounds, CoDaCoRe implements a “discretization” procedure that exploits the information learned by the soft assignment vector \( \tilde{w} \), in order to efficiently identify a pair of sparse subsets \((J^+, J^-)\).

The most straightforward way to convert the (soft) assignment \( \tilde{w} \) into a (hard) pair of subsets is by fixing a threshold \( t \in (0, 1) \):

\[
J^+ = \{ j : \tilde{w}_j > t \}, \tag{9}
\]

\[
J^- = \{ j : \tilde{w}_j < -t \}. \tag{10}
\]

Note that given a trained \( \tilde{w} \) and a fixed threshold \( t \), we can evaluate the quality of the corresponding balance \( B(x; J^+, J^-) \) (resp. amalgamation) by optimizing Eq. 4 over \((\alpha, \beta)\) alone, i.e., fitting a linear model. Computationally, fitting a linear model is much faster than optimizing Eq. 8, and can be done repeatedly for a range of values of \( t \) with little overhead. In CoDaCoRe, we combine this strategy with cross-validation in order to select the threshold, \( \hat{t} \), that optimizes predictive performance (see Section A of the Supplement for full detail). Finally, the trained regression function is:

\[
\hat{f}(x) = \hat{\alpha} + \hat{\beta} \cdot B(x; \hat{J}^+, \hat{J}^-), \tag{11}
\]

where \((\hat{J}^+, \hat{J}^-)\) are the subsets corresponding to the optimal threshold \( \hat{t} \), and \((\hat{\alpha}, \hat{\beta})\) are the coefficients obtained by regressing \( y_i \) against \( B(x_i; \hat{J}^+, \hat{J}^-) \) on the entire training set.

3.3. Regularization

Note from Equations 9 and 10 that larger values of \( t \) result in fewer covariates assigned to the balance \( B(x; \hat{J}^+, \hat{J}^-) \), i.e., a sparser model. Thus, CoDaCoRe can be regularized simply by making \( \hat{t} \) larger. Similarly to lasso regression, our implementation of CoDaCoRe uses the 1-standard-error rule: namely, to pick the sparsest model (i.e., the highest \( \hat{t} \)) with mean cross-validated score within 1 standard error of the optimum (Friedman et al., 2001). Trivially, this rule can be generalized to a \( \lambda \)-standard-error rule, where \( \lambda \) becomes a regularization hyperparameter that can be tuned by the practitioner if so desired (with lower values trading off some sparsity in exchange for predictive accuracy). For consistency, we restrict our experiments to \( \lambda = 1 \), however our results can be improved further by tuning \( \lambda \) on each dataset.

In practice, we recommend choosing a lower value (e.g., \( \lambda = 0 \)) when the emphasis is on predictive accuracy rather than interpretability or sparsity, though our benchmarks still show competitive performance with the choice of \( \lambda = 1 \).

Algorithm 1 CoDaCoRe

**Inputs:** Training data: \( (x_i, y_i)_{i=1}^n \).

Initialize \( \hat{g}(x) = 0 \).

repeat

- Initialize a new relaxation \((w, \alpha, \beta)\).
- Train \((w, \alpha, \beta)\) by gradient descent.
- Use cross-validation to find the optimal threshold, \( \hat{t} \).
- Retrain \((\alpha, \beta)\) using \((\hat{J}^+, \hat{J}^-)\).
- Update ensemble \( \hat{g}(x) \leftarrow \hat{g}(x) + \hat{f}(x) \).

until \( \hat{J}^+ = \emptyset \) or \( \hat{J}^- = \emptyset \).

Return \( \hat{g}(x) \).

3.4. CoDaCoRe Algorithm

The computational efficiency of our continuous relaxation allows us to train multiple regressors of the form of Eq. 11 within a single model. In the full CoDaCoRe algorithm, we ensemble multiple such regressors in a stage-wise additive fashion, where each successive balance is fitted on the residual from the current model. Thus, CoDaCoRe identifies a sequence of balances, in decreasing order of importance, each of which is sparse and interpretable. Training terminates when an additional relaxation (Eq. 6) cannot improve the cross-validation score relative to the existing ensemble (equivalently, when we obtain \( \hat{t} = 1 \)). Typically, only a small number of balances is required to capture the signal in the data, and as a result CoDaCoRe produces very sparse models overall, further enhancing interpretability. Our procedure is summarized in Algorithm 1.

3.5. Amalgamations

CoDaCoRe can be used to learn amalgamations (Eq. 2) much in the same way as for balances (the choice of which to use depending on the goals of the biologist). In this case, our relaxation is defined as:

\[
\hat{A}(x_i; w) = \log \left( \frac{\sum_{j} \tilde{w}_j^+ x_{ij}}{\sum_{j} \tilde{w}_j^- x_{ij}} \right) \tag{12}
\]

\[
= \log \left( \frac{\tilde{w}^+ \cdot x_i}{\tilde{w}^- \cdot x_i} \right), \tag{13}
\]

i.e., we approximate summations over subsets of the inputs, with weighted summations over all components (compare Eq. 2 and Eq. 12). The rest of the argument follows verbatim, replacing \( B(\cdot) \) with \( A(\cdot) \) and \( \hat{B}(\cdot) \) with \( \hat{A}(\cdot) \) in Equations 3, 4, 8, and 11.

3.6. Extensions

Our model allows for a number of extensions:

- **Unsupervised learning.** By means of a suitable unsupervised loss function, CoDaCoRe can be extended to unlabelled datasets, \( (x_i)_{i=1}^n \), as a method for identi-
Learning Sparse Log-Ratios for High-Throughput Sequencing Data

fying log-ratios that provide a useful low-dimensional representation. Such a method would automatically provide a scalable alternative to several existing dimensionality reduction techniques for CoDa (Pawlowsky-Glahn et al., 2011; Mert et al., 2015; Martín-Fernández et al., 2018; Greenacre, 2019b; Martino et al., 2019).

• Incorporating confounders. In addition to \((x_i, y_i)_{i=1}^n\), in some applications the effect of additional (non-compositional) predictors, \(z_i\), is also of interest. In this case, the effect of \(z_i\) can be “partialled out” a priori by first regressing \(y_i\) on \(z_i\) alone, and using this regression as the initialization of the CoDaCoRe ensemble. Alternatively, \(z_i\) can also be modeled jointly in Equations 3 and 11 (e.g., by adding a linear term \(\gamma \cdot z_i\)) (Forslund et al., 2015; Noguer-Julian et al., 2016; Rivera-Pinto et al., 2018).

• Nonlinear regression functions. Our method extends naturally to nonlinear regression functions of the form \(f(x) = h_\theta(B(x; J^+, J^-))\), where \(h_\theta\) is a parameterized differentiable family. These functions include neural networks, which have recently become of interest in microbiome research (Morton et al., 2019a; Quinn et al., 2020).

• Applications to non-compositional data. Aggregations of parts can be useful outside the realm of CoDa; for example, an amalgamation applied to a categorical variable with many levels represents a grouping of the categories (Bondell & Reich, 2009; Gertheiss & Tutz, 2010; Tutz & Gertheiss, 2016).

4. Experiments

We evaluate CoDaCoRe on a collection of 25 benchmark datasets including 13 datasets from the Microbiome Learning Repo (Vangay et al., 2019), and 12 microbiome, metabolite, and microRNA datasets curated by Quinn & Erb (2019). These data vary in dimension from 48 to 3,090 covariates (see Section B of the Supplement for a full description). For each dataset, we fit CoDaCoRe on 20 random 80/20 train/test splits, sampled with stratification by case-control (He & Ma, 2013). We compare against:

• Interpretable models (Sections 2.1 and 2.2); pairwise log-ratios (Greenacre, 2019b)
• selbal (Rivera-Pinto et al., 2018), and amalgam (Quinn & Erb, 2020). We also consider lasso logistic regression (with regularization parameter chosen by cross-validation with the 1-standard-error rule).

• Other CoDa models (Section 2.3): Coda-lasso (Lu et al., 2019), DeepCoDA (Quinn et al., 2020), and Susin et al. (2020). Note that these methods learn (weighted) geometric averages over a large number of input variables, which are evidently not as straightforward to interpret as simple balances or amalgamations.

• Black box classifiers: Random Forest and XGBoost, where we tune the model complexity parameters by cross-validation (subsample size and early stopping, respectively).

4.1. Results

We evaluate the quality of our models across the following criteria: computational efficiency (as measured by runtime), sparsity (as measured by the percentage of input variables that are active in the model), and predictive accuracy (as measured by out-of-sample accuracy and ROC AUC). Table 2 provides an aggregated summary of the results; CoDaCoRe (with balances) is the only method that scales effectively to our larger datasets, while consistently achieving high predictive accuracy. Moreover, its performance is broadly consistent across smaller and larger datasets.

Figure 1 shows the average runtime of our classifiers on each dataset, with larger points denoting larger datasets. CoDaCoRe trains orders of magnitude faster and scales better than existing interpretable CoDa methods. On our larger datasets (3,090 inputs), selbal runs in ~100 hours, pairwise log-ratios and amalgam both run in ~10 hours, and CoDaCoRe runs in under 10 seconds (full runtimes are provided in Table 2 in the Supplement). All runs, including those involving gradient descent, were performed on identical CPU...
cores; CoDaCoRe can be accelerated further using GPUs, but we did not find it necessary to do so. It is also worth noting that the outperformance of CoDaCoRe is not merely as a result of the other methods failing on high-dimensional datasets. The consistent performance of CoDaCoRe across smaller and larger datasets is demonstrated in Supplementary Tables 3, 4, and 5, which show a full breakdown of results across each dataset.

Not only is CoDaCoRe sparser and more accurate than other interpretable models, it also performs on par with state-of-the-art black-box classifiers. By simply reducing the regularization parameter, from $\lambda = 1$ to $\lambda = 0$, CoDaCoRe (with balances) achieved an average 77.6% out-of-sample accuracy and 82.0% AUC, on par with Random Forest and XGBoost (bottom rows of Table 2), while only using 5.9% of the input variables, on average. This result indicates, first, that CoDaCoRe provides a highly effective algorithm for variable selection in high-dimensional HTS data. Second, the fact that CoDaCoRe achieves similar predictive accuracy as best-in-class black-box classifiers, suggests that our model may have captured a near-complete representation of the signal in the data. At any rate, we take this as evidence that log-ratio transformed features are indeed of biological importance in the context of HTS data, corroborating previous microbiome research (Rahat-Rozenbloom et al., 2014; Crovesy et al., 2020; Magne et al., 2020).

### 4.2. Interpretability

The CoDaCoRe algorithm offers two kinds of interpretability. First, it provides the analyst with sets of covariates whose aggregated ratio predicts the outcome of interest. These sets are easy to understand because they are discrete, with each component making an equivalent (unweighted) contribution. They are also sparse, usually containing fewer than 10 features per ratio, and can be made sparser by adjusting the regularization parameter $\lambda$. Such ratios have a precedent in microbiome research, for example the Firmicutes-to-Bacteroidetes ratio is used as a biomarker of gut health (Crovesy et al., 2020; Magne et al., 2020). Second, CoDaCoRe ranks predictive ratios hierarchically. Due to the ensembling procedure, the first ratio learned is the most predictive, the second ratio predicts the residual from the first, and so forth. Like principal components, the balances (or amalgamations) learned by CoDaCoRe are naturally ordered in terms of their explanatory power. This ordering aids interpretability by decomposing a multivariable model into comprehensible “chunks” of information.

Notably, we find a high degree of stability in the log-ratios selected by the model. We repeated CoDaCoRe on 10 independent training set splits of the Crohn disease data provided by Rivera-Pinto et al. (2018), and found consensus among the learned models. Figure 2 shows which bacteria were included for each split, in both versions of CoDaCoRe (balances and amalgamations). Importantly, most of the bacteria that were selected consistently by CoDaCoRe – notably Dialister, Roseburia and Clostridiales – were also identified by Rivera-Pinto et al. (2018). Differences between the sets selected by CoDaCoRe with balances vs. CoDaCoRe with amalgamations can be explained by differences in how the geometric mean vs. summation operations impact the log-ratio. The geometric mean, being more sensitive to small numbers, is more affected by the presence of rarer bacteria species like Dialister and Roseburia (as compared with the more common bacteria species like Faecalibacterium and Faecalibacterium).

### 4.3. Scaling to Liquid Biopsy Data

HTS data generated from from clinical blood samples can be described as a “liquid biopsy” that can be used for cancer diagnosis and surveillance (Best et al., 2015; Alix-Panabières, 2020).
Learning Sparse Log-Ratios for High-Throughput Sequencing Data

CoDaCoRe - Balances

Dialister
Aggregatibacter
Lactobacillales
Streptococcus
Parabacteroides
Peptostreptococcaceae
Faecalibacterium
Lachnospiraceae
Clostridiales
Roseburia

Figure 2. CoDaCoRe variable selection for the first (most explanatory) log-ratio on the Crohn disease data (Rivera-Pinto et al., 2018). For each of 10 independent training set splits (80% of the data), we show which variables are selected in the numerator (blue) and denominator (orange) of the log-ratio. Both versions of CoDaCoRe, with balances (top) or amalgamations (bottom), learn remarkably consistent log-ratios across independent training sets.

Table 3. Evaluation metrics for the liquid biopsy data (Best et al., 2015), averaged over 20 independent 80/20 train/test splits. CoDaCoRe (with balances) achieves equal predictive accuracy as competing methods, but with much sparser solutions. Note that sparsity is expressed as an (integer) number of active variables in the model (not as a percentage of the total, as was done in Table 1). Running time is shown in seconds (standard errors were small and are omitted for brevity).

<table>
<thead>
<tr>
<th>Time</th>
<th># Vars</th>
<th>ACC. (%)</th>
<th>AUC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASELINE</td>
<td>0</td>
<td>0±0.0</td>
<td>79.1±0.0</td>
</tr>
<tr>
<td>CoDaCoRe</td>
<td>31</td>
<td>3±1</td>
<td>91.0±1.9</td>
</tr>
<tr>
<td>LASSO</td>
<td>23</td>
<td>22±4</td>
<td>87.8±1.3</td>
</tr>
<tr>
<td>RF</td>
<td>383</td>
<td>-</td>
<td>89.0±1.6</td>
</tr>
<tr>
<td>XGBOOST</td>
<td>108</td>
<td>-</td>
<td>90.6±1.9</td>
</tr>
</tbody>
</table>

These data can be fitted to these data at a similar computational cost to a single lasso regression, i.e., under a minute on a single CPU core. Thus, CoDaCoRe can be used to discover interpretable and predictive log-ratios that are suitable for liquid biopsy cancer diagnostics, among other similar applications.

We showcase the capabilities of CoDaCoRe in this high-dimensional setting, by applying our algorithm to the liquid biopsy data of (Best et al., 2015). These contain \( p = 58,037 \) genes sequenced in \( n = 288 \) human subjects, 60 of whom were healthy controls, the others having been previously diagnosed with cancer. Averaging over 20 random 80/20 train/test splits of this dataset, we found that CoDaCoRe achieved the same predictive accuracy as competing methods (within error), but obtained a much sparser model. Remarkably, CoDaCoRe identified log-ratios involving just 3 genes, that were equally predictive to both black-box classifiers and linear models with over 20 covariates. This case study again illustrates the potential of CoDaCoRe to derive novel biological insights, and also to develop learning algorithms for cancer diagnosis, a domain in which model interpretability – including sparsity – is of paramount importance (Wan et al., 2017).

5. Conclusion

Our results corroborate the summary in Table 1: CoDaCoRe is the first sparse and interpretable CoDa model that can scale to high-dimensional HTS data. It does so convincingly, with linear scaling that results in runtimes similar to linear models. Our method is also competitive in terms of predictive accuracy, performing comparably to powerful black-box classifiers, but with interpretability. Our findings suggest that CoDaCoRe could play a significant role in the future analysis of high-throughput sequencing data, with broad implications in microbiology, statistical genetics, and more generally, in the field of CoDa.
Learning Sparse Log-Ratios for High-Throughput Sequencing Data


Learning Sparse Log-Ratios for High-Throughput Sequencing Data


Greenacre, M. Amalgamations are valid in compositional data analysis, can be used in agglomerative clustering, and their logratios have an inverse transformation. *Applied Computing and Geosciences*, 5:100017, 2020.


He, H. and Ma, Y. Imbalanced learning: foundations, algorithms, and applications. 2013.


Learning Sparse Log-Ratios for High-Throughput Sequencing Data


Learning Sparse Log-Ratios for High-Throughput Sequencing Data