A Comparison of Dimensionality Reduction Methods for Large Biological Data

ASHLEY BABJAC, TAYLOR ROYALTY, ANDREW D STEEN, and SCOTT J EMRICH, University of Tennessee, Knoxville, USA

Large-scale data often suffer from the curse of dimensionality and the constraints associated with it; therefore, dimensionality reduction methods are often performed prior to most machine learning pipelines. In this paper, we directly compare autoencoders performance as a dimensionality reduction technique (via the latent space) to other established methods: PCA, LASSO, and t-SNE. To do so, we use four distinct datasets that vary in the types of features, metadata, labels, and size to robustly compare different methods. We test prediction capability using both Support Vector Machines (SVM) and Random Forests (RF). Significantly, we conclude that autoencoders are an equivalent dimensionality reduction architecture to the previously established methods, and often outperform them in both prediction accuracy and time/memory performance when condensing large, sparse datasets.

CCS Concepts: • **Computing methodologies** → *Feature selection*; Cross-validation; **Supervised learning by classification**; **Learning latent representations**; Classification and regression trees; Support vector machines.

Additional Key Words and Phrases: Autoencoders, LASSO, PCA, manifold learning, dimensionality reduction, SVM, Random Forests

ACM Reference Format:
Ashley Babjac, Taylor Royalty, Andrew D Steen, and Scott J Emrich. 2022. A Comparison of Dimensionality Reduction Methods for Large Biological Data. 1, 1 (May 2022), 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Dimensionality reduction is a common task in most machine learning frameworks. When dealing with extremely large data it is imperative to reduce noise and other confounding factors that can either complicate or introduce error when training a model. In bioinformatics, this is traditionally approached using one of two methods: Principal Components Analysis (PCA) [1] or LASSO (Least Absolute Shrinkage and Selection Operator) [8]. A primary reason for their general acceptance by the biological research community is their ability to be easily interpreted.

More recently, deep learning methods such as autoencoders and transformers have been making their way into more bioinformatics applications. These models have been recognized as being able to extract signal from hard to model datasets using a wide variety of features including available metadata. One such example of this was using autoencoders for predicting microbiomes from environmental factors [9]. In another more recent example, Google's BERT transformer model was used to perform text analysis on DNA sequences [13]. We continue this trend by applying autoencoders for dimensionality reduction on difficult to model microbial and microbiome data.

Authors' address: Ashley Babjac, ababjac@vols.utk.edu; Taylor Royalty, troyalty@vols.utk.edu; Andrew D Steen, asteen1@utk.edu; Scott J Emrich, semrich@utk.edu, University of Tennessee, Knoxville, 2704 Kingston Pike, Knoxville, Tennessee, USA, 37919-4618.

1.1 Related Work

Similar work has been done in trying to better understand biological associations. One such tool is MetAML that pairs feature selection (LASSO or Elastic Net) with machine learning classifiers (Support Vector Machines (SVM) and Random Forests (RF)) [21]. Their results show that combining feature selection with a machine learning classifier significantly improves performance for metagenomic prediction from microbiomes. The unnamed approach of [9] is similar to our approach in that they use autoencoders to develop an initial latent space based on species presence/absences; however, their focus was using environmental data as the only input for predicting the composition of root microbiomes. No traditional feature selection or ordination was considered, and only maize (corn) data was used to train and assess their framework. Mian [14] is a newly available web-based microbiome visualization and machine learning platform. In addition to regression models used by [9] and similar classifiers, they provide customizable multi-layer perceptrons (MLPs). An MLP-based approach typically did worse on maize root microbiome data than an autoencoder-based approach [9] and, in addition, Mian only uses simple dimensionality reduction such as univariate feature selection. In our prior work, we showed that single feature-based selection does not work as well as LASSO-based feature selection on the most challenging data considered here [24]. Moreover, autoencoders have been shown to perform well in dimensionality reduction tasks [25, 29, 30].

1.2 Our Approach and Contributions

In this paper we explicitly compare pseudo-dimensionality reduction using autoencoders to the more traditionally accepted methods of PCA and LASSO as well as to a common manifold learning technique (t-SNE). Specifically, we use an autoencoders' latent space representation in combination with classification algorithms, similar to how MetAML [21] pairs either elastic net or LASSO-based feature selection with classification. We compare the feature selection approaches with and without additional metadata, and assess the resulting prediction performance.

The primary contributions of this work are: (i) developing a model architecture for using autoencoders and machine learning classification for improved performance across multiple hard to model datasets with different use cases, (ii) using this model architecture to improve speed/memory consumption when performing prediction tasks on very large data. We conclude based on these results that autoencoder-based dimensionality reduction outperforms previously used PCA, LASSO and manifold learning frameworks in terms of both classification performance and runtime.

2 MATERIALS AND METHODS

2.1 Data

We use four different datasets for comparing dimensionality reduction and classification. Each dataset consists of what we define as "metadata" (features that describe information about each sample), and "features" (the actual features for each sample). We will define the specifics for each of these data / use cases below.

The first dataset we use comes from the TARA oceans project [27], and is comprised of 124 samples by 9,305 features, 36 of which are metadata. For these data, the metadata are variables about the ocean sample (i.e., depth, pressure, etc.) and the primary features are the observed gene abundances at each site grouped based on individual Kegg Ontology terms. We predict the ocean region where each sample was collected and refer to these data as "TARA" throughout. It is important to note that the TARA labels (9 ocean regions total) are highly unbalanced. Since TARA has such a small number of samples we perform SMOTE (synthetic minority-oversampling technique) [4] to re-balance the data during the pre-processing step of our pipeline (see Figure 1), but we note the ocean region label has very strong signal

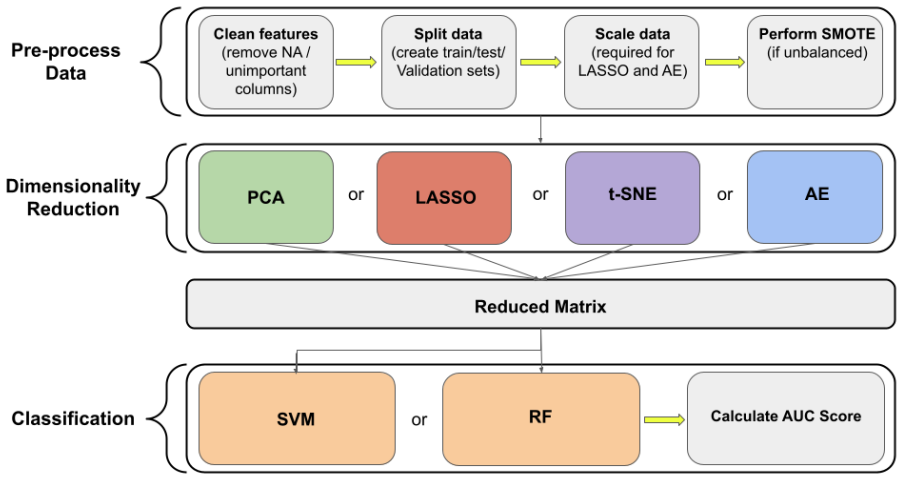

Fig. 1. Our model pipeline for computing desired labels via dimensionality reduction and SVM.

even without re-balancing. Hence, these data act as a good control for verifying our results because we expect good performance relatively independent from which model we choose.

The second dataset comes from soil rhizosphere samples [15] (and will be referred to as "Rhizo"). It is made up of 62 samples by 13,822 features, two of which are treated as metadata (habitat, irrigation). The features themselves are sparse Amplicon Sequence Variant (ASV) abundances from a structured experimental design where two different poplar (*Populus trichocarpa*) genotypes with either high or low drought tolerance, respectively, are grown in a common garden under two different irrigation schemes: normal or reduced. This an especially difficult prediction task because drought tolerance is defined based on the host poplar tree genotype and not an individual soil rhizosphere sample. We previously analyzed these data using LASSO in [24]. These should be challenging data for a deep learning-inspired framework as the time and resource-intensive nature of this common garden experiment yielded at most 16 samples per genotype/treatment, e.g., drought tolerant with reduced irrigation. It should be ideal for considering metadata, however, because of the confounding factor of irrigation in the experimental design.

The third set of data comes from a reanalysis of public malaria expression data obtained in a clinic from infected patients [31]. It contains 1,045 samples by 5,081 features, 22 of which are metadata. The metadata contains information about each patient and sample (ex: location, temperature of patient, time, drug used for treatment, etc.), and the features are the normalized expression values for each gene relative to a common lab strain (3D7). We predict the label "resistant", where a 1 represents a clearance value >= 6 (resistant), and a 0 represents a clearance value < 6 (not resistant). Similar to TARA, the labels for these data are highly unbalanced and SMOTE is performed before prediction.

The final dataset is generated from GEM (Genomes from Earth's Microbiomes) project data, which consists of metagenome-assembled genomes assembled from a diverse set of environments [18]. For background, some microbes can be grown in culture very easily, but most cells on Earth belong to species that resist growing in pure culture [17]. It is likely that there is a biological reason that many microbial species do not grow well under standard culturing conditions, e.g., slow growth or extreme sensitivity to specific chemicals [10, 12]. As a step towards identifying this reason, we compared taxonomic labels of GEM genomes to those of genomes present in RefSeq, a database of genomes of microbes

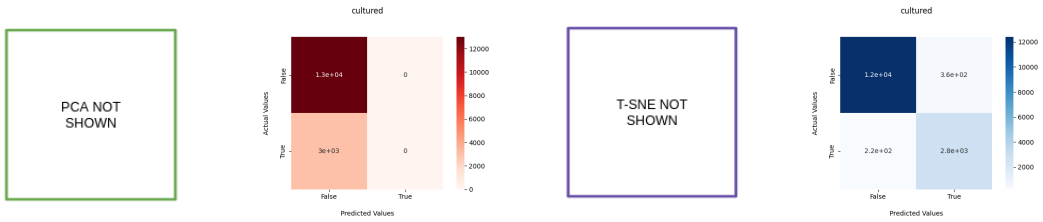

Fig. 2. Comparing LASSO (red) and autoencoders (blue) results using SVM classification on GEM feature vectors containing no metadata. We note that the confusion matrices for PCA and t-SNE are not visualized here because they never completed running in >2 weeks. Note that the LASSO-based model predicts all samples are uncultured.

that have been grown in pure culture [19]. GEM genomes were assigned via the Genome Taxonomy Database [20] a label of "cultured" when their taxonomic inference was present in the RefSeq database, or "uncultured" if not.

This custom GEM data generated by us for this attempt at modeling culturability contains 52,515 samples by 4,389 features. The features also include 12 taxonomy related metadata from the mapping to RefSeq, i.e., inferred phylum, class, genus, species, taxonomic distance to a known cultured organism. We predict the cultured status as described above using either 71 pathway features that represent coarse-grained COG assignments (aggregated into higher biochemical-pathways) or 4,318 annotation features that are more fine-grained COG assignments. The value of each feature is the approximate number of genes in a given genome/species with a specific COG assignment. These GEM data are an ideal input for an autoencoder-based framework given prior work [9] because: (i) We have tens of thousands of samples (genomes); and (ii) more importantly, **there is no known model that can predict culturability available to microbial ecologists**. There have been prior efforts to associate microbiomes with diseases (e.g., [16, 21]) or specific plants (e.g., [6]) but to the best of our knowledge no one has trained a model using genes/pathways within specific organisms to predict cryptic phenotypes such as culturability.

2.2 Models for Dimensionality Reduction

We evaluate four different models for dimensionality reduction: PCA, LASSO, manifold learning (t-SNE) and autoencoders.

PCA (principal component analysis) [1] is a model that works by projecting data points into hyperplanes that best approximate the original data via least squares error. PCA can either project into a K dimensional space (i.e. create K components to represent the data), or it can account for K% of variability (by creating N components). For our model, we chose to account for 90% of variability based on preliminary testing. This amount of variability created a small enough number of components to be comparable (and fast) while still accounting for the majority of variability in the data across all of the datasets considered here.

LASSO [8] is a modified form of linear regression that uses a cost function to tune the model based on an α hyperparameter. The lower the α parameter, the more LASSO begins to approximate linear regression, whereas higher α values tend to cause coefficients to trend towards 0. Unlike PCA and autoencoders, where we encode the entire dataset into a lower dimensional space, LASSO reduces (or increases) feature importance based on the determined regression co-efficients. For our model, we tested α values in range (1, 10, 0.1). The best value of α was dataset dependent and was used in each run to create the matrix passed to each classification model based on 5-fold cross validation. The initial

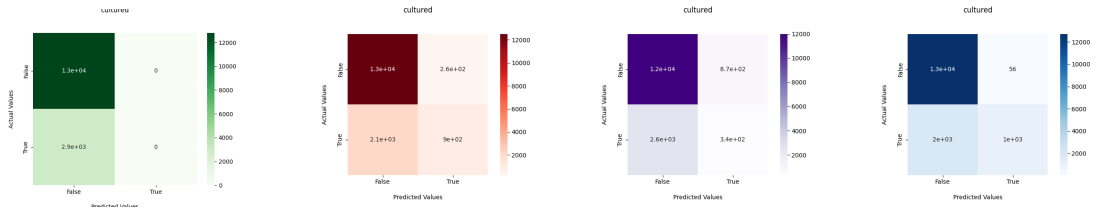

Fig. 3. Comparing PCA (green), LASSO (red), t-SNE (purple) and autoencoders (blue) results using Random Forests for classification with feature vectors containing no metadata from GEM.

range was chosen based on preliminary testing where α values lower than 1 tended to not perform as well when their results were passed to the classification task.

Manifold Learning techniques [3] are often thought of as an unsupervised, non-linear equivalent of linear projection methods such as PCA. These algorithms are often used for visualization purposes. There are many different manifold learning techniques, but we specifically work with t-SNE (t-Distributed Stochastic Neighbor Embedding) [28], which is often used in analyzing RNAseq data. T-SNE converts "tendencies" of data points to probabilities based on trends. The tendencies in the original space are represented by Gaussian joint probabilities and then embedded using Student's t-distributions. Since t-SNE is heavily computationally intensive, we use the Barnes-Hut approximation strategy, with the number of components set to 3 (the max allowed). We note that since t-SNE is generally used for visualization purposes, the number of components is often set to 2; however, nc=3 had higher AUC across all datasets tested here.

Autoencoders [2] are a type of neural network that involve an input layer (referred to as the encoder) and an output layer (referred to as the decoder). Autoencoders, in general, condense the original input data down to a small number of layers (referred to as the latent space) and then try to recreate that input from the latent space representation. For our model, we use the latent space representation as the reduced matrix to pass to the classification task. There are multiple hyperparameters that can be tuned for autoencoders. We find the best combination by varying the following parameters via grid search with 5-fold cross validation and training of 10 epochs: {latent dimensions: [10, 50, 100, 200], activation function: [relu, sigmoid, tanh], loss: [MAE, binary crossentropy], optimizer: [SGD, Adam]}. We use the best combination of parameters to encode the latent space for each dataset prior to classification.

2.3 Architecture

Dimensionality reduction does not perform classification; therefore, we pass the dimensionality reduced feature vectors through SVM and Random Forests to perform classification on our labels and assess the results. Figure 1 shows our model architecture. The specifics of our pipeline are as follows: (i) pre-process the data; this involves cleaning the data to remove any unknown values, calling SMOTE if required on unbalanced data, and scaling the data based on model requirements, (ii) perform dimensionality reduction using PCA, LASSO, t-SNE or autoencoders as described in section 2.2; at the end of this step a reduced matrix is created based on the specified algorithm, (iii) perform SVM or Random Forests based on the reduced matrix; this is called with probabilities set to true in order to compute the AUC and plot the ROC curve. We use SVM and Random Forests for classification based on the pipeline previously established by metAML [21]. Specifically, SVM is tuned with a grid search of the following parameters: {C value: [0.1, 1, 10, 100, 1000], gamma: [1, 0.1, 0.01, 0.001, 0.0001], kernel: [rbf, linear]}. Similarly, Random Forests are tuned with grid search and standard associated parameters: {number of estimators: [200, 500], max features: [auto, log2, sqrt], max depth: [4,5,6,7,8],

Dataset	TARA				Rhizo				Malaria				GEM			
Model	PCA	LASSO	t-SNE	AE	PCA	LASSO	t-SNE	AE	PCA	LASSO	t-SNE	AE	PCA	LASSO	t-SNE	AE
With Metadata	**0.997**	0.984	0.523	0.996	**0.530**	0.500	0.500	0.500	0.960	**0.980**	0.410	0.700	N/A	**1.000**	N/A	0.980
Without Metadata	0.990	0.976	0.536	**0.994**	0.470	0.500	**0.680**	0.590	**0.760**	0.510	0.540	0.640	N/A	0.700	N/A	**0.960**

Table 1. Table showing the AUC scores for each combination of dimensionality reduction and data using SVM for prediction. The TARA AUC score is averaged across the 9 site prediction scores. The bolded scores represent the best performing model for each dataset with and without metadata. PCA and t-SNE are shown as N/A because they did not complete (>2 weeks running).

Dataset	TARA				Rhizo				Malaria				GEM			
Model	PCA	LASSO	t-SNE	AE	PCA	LASSO	t-SNE	AE	PCA	LASSO	t-SNE	AE	PCA	LASSO	t-SNE	AE
With Metadata	0.979	**0.985**	0.492	0.984	0.620	**0.700**	0.530	0.500	0.920	**1.000**	0.600	0.540	0.630	**1.000**	0.580	0.980
Omit Metadata	0.908	0.986	0.581	**0.989**	0.580	**0.680**	0.390	0.530	0.540	**0.620**	0.480	0.590	0.630	0.850	0.680	**0.920**

Table 2. Table showing the AUC scores for each combination of dimensionality reduction and data using Random Forests for prediction. The TARA AUC score is averaged across the 9 site prediction scores. The bolded scores represent the best performing model for each dataset with and without metadata.

criterion: [gini, entropy]}. We perform 5-fold cross validation and parallelize the classification computations across 5 cores. We do note that this architecture is not necessarily limited to classification, and could easily be applied to other models besides SVM and Random Forest. Further, all of our code and figures from the full analysis are available on GitHub at ababjac/microbial-steen-project and are built off of scikit-learn [23] and Keras [5] model API's.

2.4 Metrics for Evaluation

To evaluate our models we look at both prediction performance and speed/memory consumption (reported as wall clock time and total allocated memory in MiB) during runtime. The wall clock time is calculated using the standard /usr/bin/time on Linux and the memory is profiled at each timestep (0.1 seconds) using the Python "mprof" command line tool [22]. To evaluate prediction we use AUC (area under the curve) [11], which is a number between 0 and 1 calculated as the area underneath the ROC curve (showing true vs false positive rate across all classification thresholds). For balanced data, we would expect a minimum AUC of 0.5 if we predicted everything as true for a baseline comparison. Because it is known that the metadata features contain the bulk of important information correlated with the prediction labels, we perform the analysis described in Section 2.3 both including and excluding the metadata features for each dataset in order to compare dimensionality reduction methods when given data with low signal.

3 RESULTS

3.1 Autoencoders produce the best condensed representation when metadata features are omitted

Tables 1 and 2 show the AUC scores produced from running our model architecture across all three datasets. An autoencoder combined with SVM classification has the best performance when metadata are not included for the ecological data (TARA and GEM). Although t-SNE is the best performing model for the host-synmbiont data, this method performs extremely poorly in all other cases. For the Malaria data, PCA performs the best with SVM (AUC=0.76), but autoencoders are the second best model (AUC=0.64).

The robust and consistent performance of autoencoders is significant because (i) we are able to extract strong signal using deep learning methods from datasets even if they have few samples due to experimental constraints, and (ii) we are able to predict solely based on hard to understand data with no external context. The latter is further supported by predicting with random forests (RF) instead of SVM classification (Table 2). We can see that for both TARA and

Dataset	TARA				Rhizo				Malaria				GEM			
Model	PCA	LASSO	t-SNE	AE	PCA	LASSO	t-SNE	AE	PCA	LASSO	t-SNE	AE	PCA	LASSO	t-SNE	AE
Time(SVM)	0:1:7	0:34:37	19:19:23	1:49:31	0:0:12	0:6:29	2:37:29	0:14:12	0:0:22	0:3:27	8:4:24	0:10:25	>2 weeks	1 day	>2 weeks	12:18:42
Mem(SVM)	266.0	27,332.2	513.5	26,210.1	226.7	6,120.3	214.4	12,134.8	456.9	24,205.7	408.7	6,019.6	N/A	12,897,319.0	N/A	6,126,707.7
Time(RF)	0:5:8	0:29:26	0:13:52	1:8:18	0:0:39	0:4:24	0:1:34	0:7:17	0:2:12	0:3:49	0:1:21	0:11:14	0:17:56	2:36:18	0:20:50	6:34:29
Mem(RF)	238.4	28,702.4	559.7	37,367.0	227.3	5,828.5	212.7	11,873.4	459.8	24,361.2	386.1	6,165.9	143,005.1	12,939,477.5	224,169.6	10,089,669.3

Table 3. Table showing the timing for all combinations of models. The statistics are averaged across 5 runs using our pipeline with no metadata. The timing is the elapsed time formatted as hours:minutes:seconds, rounded up to the nearest second. The memory is the sum of increases in allocated memory at each timestep (every 0.1 seconds sampled) in MiB.

GEM autoencoders again have the highest AUC when predicting without metadata, but the others seem to be best with LASSO and Random Forest. Since Rhizo is such a small dataset (62 samples), it is unsurprising that autoencoders have trouble extracting signal without more data. Autoencoders suffer in training with the Malaria data due to the unbalanced labels in the training set. For GEM where we have a large amount of data (and balanced samples), we see an increase of 46% over PCA, 8% over LASSO, and 41% over t-SNE when predicting with Random Forest (see Figure 3).

3.2 Traditional feature selection methods garner slightly more prediction power when using metadata

PCA and LASSO tend to slightly outperform autoencoders when metadata features are included in building the model. Looking at Table 1, we can observe that when predicting with SVM the best performing models are PCA (0.997 for TARA), PCA (0.53 for Rhizo) and LASSO (1.00 for GEM). While autoencoders do not perform the best in any case, we do note that they are consistently the second best model (with the exception of the Malaria results) and only trail behind by a few percentage points in all cases. Similarly, when predicting with Random Forests, LASSO performs the best in all three cases (0.985 for TARA, 0.700 for Rhizo, and 1.00 for GEM). Again, we see autoencoders are the second best performing model in all cases where the metadata is included, except for predicting on Rhizo due to the small number of samples and the overall weak signal in these data [15, 24].

3.3 Autoencoders can improve timing and memory consumption for very large datasets

Table 3 shows the timing and memory statistics when running our pipeline excluding metadata. For small data, PCA is consistently the fastest, followed by LASSO followed by autoencoders. Autoencoder models have significant overhead (even for small) data because of the extensive grid search combined with cross-validation (and training for n epochs for each cross validation step). However, autoencoders significantly increase time performance when combined with other machine learning methods (SVM) due to the nature of the latent space. For large data, machine learning methods can be quite time consuming (especially if the wrong parameters or kernel is chosen). Using autoencoders allows for a much more condensed representation of the data than PCA or LASSO while still retaining a majority of the information. For large data, this significantly smaller representation allows for a much improved classification runtime and outweighs the overhead of training an autoencoder. Specifically, for GEM we can see that the runtime when comparing autoencoders to LASSO is cut in half (approximately 12 hours for GEM and 1 day for LASSO), and PCA takes more than 2 weeks to run, which is not feasible for most projects. Autoencoders also have additional memory overhead, but as the data size increases, they tend to begin outperforming other methods (specifically LASSO for Malaria/GEM).

4 DISCUSSION

One key distinction between our model and previous work is looking at different types of dimensionality reduction with and without metadata features on diverse data: microbiome abundances, host-symbiont confounded by experimental

treatment, clinical gene expression, and gene content in thousands of genomes found in the environment. Across all datasets when metadata are omitted, the autoencoder model combined with SVM classification finishes at least second and outperforms the next best dimensionality reduction model by a large margin on the largest dataset (GEM). Autoencoders are therefore not always the best suited for all data. For example, autoencoders are the second best model to PCA and LASSO when predicting malaria drug resistance without metadata for SVM and Random Forest, respectively. Upon further inspection, we observed that the autoencoder often mis-predicts the majority label, which suggests it has learning bias from the originally imbalanced training set (before SMOTE). In support of this, when additional preprocessing of the data is performed to remove "close to resistant" labels (i.e. labels with clearance between 5 and 6), the training data are slightly more balanced and the resulting autoencoder performance does improve.

The other exceptions are related to the Rhizo data. When modeling with SVM, t-SNE outperforms autoencoders. This is likely due to the fact that t-SNE is actually designed to work with SVM (because it is split into 2 or 3 components which easily fit with SVM's various kernels). Since this is the only model where t-SNE performs "well", we do not consider this result particularly interesting. When modeling with Random Forests, autoencoders are outperformed by LASSO. Upon further inspection, when using LASSO on the Rhizo data, it actually cannot establish which features are important without the metadata – so it sets all coefficients to 0 and therefore all OTU features are included in the model as default. This leads to better prediction performance over autoencoders when using Random Forests because no dimensionality reduction was actually performed by LASSO on these data.

PCA similarly has trouble defining components without metadata. For example, when performing PCA on GEM, the PCA algorithm reduces all variability down to one component (100% of variability from all the abundance vectors). In turn, we are unable to perform SVM using PCA (takes far too long to run, see Table 1), and the Random Forest models perform significantly worse than both LASSO and autoencoders (Table 2). T-SNE in general performs very poorly (both with and without metadata), and is similar in its demand of computational resources to autoencoders (see Table 3). This suggests our pipeline is a better alternative to PCA, LASSO or t-SNE when modeling data that has no metadata information available, especially for large data. Additionally, since the way we perform the grid search is a "one size fits all" version (where the same parameters are tested across all models), additional dataset dependent fine-tuning would likely lead to significant increases in the performance of autoencoders in comparison to previously established methods.

We also recognize that PCA and LASSO slightly outperform autoencoders when including metadata in the model. One reason for this is that for all three of our datasets, multiple metadata features are highly correlated with the labels of interest. For example, when predicting on GEM with metadata, the LASSO model is able to achieve perfect prediction performance (AUC = 1.0, Table 1). Upon further inspection of the features, we can see that two of the features "cultured.level" (the level at which something has been cultured) and "taxonomic.dist" (the distance from another species based on taxonomy) are highly correlated with the cultured/uncultured status label that we are predicting, especially when combined with the feature "completeness" (how complete a sampled MAG is). LASSO basically picks out these features and passes them to the classifier which unsurprisingly performs well. When we remove these three features but leave the rest of the metadata, we get an AUC=0.70 for SVM and AUC=0.85 for RF which is exactly what LASSO predicts when using no metadata at all. Similary for Rhizo, LASSO is able to pick out the confounding factor, "irrigation", which is extremely helpful for downstream prediction, and for Malaria LASSO uses the features: "Duration of lag phase, and "PC90". We see that when combined with Random Forests, LASSO by far outperforms the other model for Rhizo (AUC = 0.70, Table 2), and this is consistent with previously reported results using a custom LASSO-based framework [24].

Additionally, using autoencoders on large data significantly improves performance when combined with machine learning classifiers. Referring to Table 3, we can see that autoencoders significantly outperform LASSO when combined

with SVM (12 hours versus roughly 1 day), and PCA/t-SNE are unable to finish in a 2 week timeframe. We suspect this is due to multiple reasons: (i) autoencoders produce a much more condensed representation than is possible from LASSO or PCA (i.e. LASSO and PCA can only reduce the number of features from m to k, so you still have n samples by k features, whereas autoencoders create layers and reshape the data into an overall much smaller representation); and, (ii) PCA and LASSO lose more information, which makes it harder for the classification algorithm to discern how to predict.

We also note that SVM can be quite kernel dependent. For our pipeline we used both RBF (Radial Basis Function) and a linear kernel. For both PCA and LASSO, the RBF and linear kernels took significant amounts of time to process at each cross-validation step because they could not easily discern where to split the hyperplanes given the low dimensional data. PCA/t-SNE took almost an hour at each cross-validation step for RBF and upwards of 10 hours for each linear step, LASSO took about 10 minutes for RBF and upwards of an hour for linear, and autoencoders took 1-5 minutes for both kernel functions. This significant speed-up of autoencoders at each step, outweighs the initial overhead of an extensive grid search. We do note that autoencoders overhead could be reduced in all models by replacing a grid search with a random search or Bayesian hyperparameter optimization that has shown relatively similar performance [26]. Alternatively, cross validation in SVM could be replaced with DBTC (distance between two classes) [7].

5 CONCLUSION

We have shown that autoencoders achieve highly competitive (if not better) performance in comparison to other previously established dimensionality reduction techniques. Importantly, we show that our framework can process traditionally hard to understand feature sets without requiring metadata context in the models, i.e., autoencoder performance on feature data alone does roughly as well or better without these additional features. We do note that autoencoder models do have some drawbacks (i.e. they are not interpretable, and for small data the increase in performance may not be worth the time/memory overhead); however, in the context of large datasets the benefits and performance improvement of autoencoders far outweigh the interpretability for classification tasks. For GEM specifically, a novel dataset we generated especially for this manuscript, we have used our model architecture to predict the culturability of an organism using only the biological gene data with 96% accuracy. This is a significant finding consistent with the hypothesis that there is a biological reason why many microbial species do not grow well under standard culturing conditions such as extreme sensitivity to specific chemicals. We leave as future work the building of a GEM model that allows for both improved prediction performance and interpretability on these data.

ACKNOWLEDGMENTS

AB was supported in part by the University of Tennessee. TMR and ADS were supported by DOE grant DE-SC0020369.

REFERENCES

[1] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* 2, 4 (2010), 433–459.
[2] Pierre Baldi. 2012. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. 37–49.
[3] Lawrence Cayton. 2005. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep* 12, 1-17 (2005), 1.
[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.
[5] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.
[6] Zhiyu Deng, Jinming Zhang, Junya Li, and Xiujun Zhang. 2021. Application of deep learning in plant–microbiota association analysis. *Frontiers in Genetics* 12 (2021).

[7] Edson Duarte and Jacques Wainer. 2017. Empirical comparison of cross-validation and internal metrics for tuning SVM hyperparameters. *Pattern Recognition Letters* 88 (2017), 6–11.

[8] Valeria Fonti and Eduard Belitser. 2017. Feature selection using LASSO. *VU Amsterdam Research Paper in Business Analytics* 30 (2017), 1–25.

[9] Beatriz García-Jiménez, Jorge Muñoz, Sara Cabello, Joaquín Medina, and Mark D Wilkinson. 2021. Predicting microbiomes through a deep latent space. *Bioinformatics* 37, 10 (2021), 1444–1451.

[10] Michael W Henson, David M Pitre, Jessica Lee Weckhorst, V Celeste Lanclos, Austen T Webber, and J Cameron Thrash. 2016. Artificial seawater media facilitate cultivating members of the microbial majority from the Gulf of Mexico. *MSphere* 1, 2 (2016), e00028–16.

[11] Jin Huang and Charles X Ling. 2005. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 17, 3 (2005), 299–310.

[12] Hiroyuki Imachi, Masaru K Nobu, Nozomi Nakahara, Yuki Morono, Miyuki Ogawara, Yoshihiro Takaki, Yoshinori Takano, Katsuyuki Uematsu, Tetsuro Ikuta, Motoo Ito, et al. 2020. Isolation of an archaeon at the prokaryote–eukaryote interface. *Nature* 577, 7791 (2020), 519–525.

[13] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. 2021. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* 37, 15 (2021), 2112–2120.

[14] Boyang Tom Jin, Feng Xu, Raymond T Ng, and James C Hogg. 2021. Mian: interactive web-based microbiome data table visualization and machine learning platform. *Bioinformatics* 38, 4 (2021), 1176–1178.

[15] Brandon Kristy, Alyssa Carrell, Eric Johnston, Jonathan Cumming, Dawn Klingeman, Kimberly Gwinn, Kimberly Syring, Caroline Skalla, Scott J. Emrich, and Melissa A. Cregger. 2022. Chronic drought differentially alters the belowground microbiome of drought tolerant and drought susceptible genotypes of *Populus trichocarpa*. *Phytobiomes* (2022), in revision.

[16] Seung Jae Lee and Mina Rho. 2022. Multimodal deep learning applied to classify healthy and disease states of human microbiome. *Scientific Reports* 12, 1 (2022), 824.

[17] Karen G. Lloyd, Andrew D. Steen, Joshua Ladau, Junqi Yin, and Lonnie Crosby. 2018. Phylogenetically novel uncultured microbial cells dominate Earth microbiomes. *mSystems* 3, 5 (sep 2018), e00055–18.

[18] Stephen Nayfach, Simon Roux, Rekha Seshadri, Daniel Udwary, Neha Varghese, Frederik Schulz, Dongying Wu, David Paez-Espino, I-Min Chen, Marcel Huntemann, et al. 2021. A genomic catalog of Earth's microbiomes. *Nature biotechnology* 39, 4 (2021), 499–509.

[19] Nuala A O'Leary, Mathew W Wright, J Rodney Brister, Stacy Ciufo, Diana Haddad, Rich McVeigh, Bhanu Rajput, Barbara Robbertse, Brian Smith-White, Danso Ako-Adjei, et al. 2016. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic acids research* 44, D1 (2016), D733–D745.

[20] Donovan H Parks, Maria Chuvochina, David W Waite, Christian Rinke, Adam Skarshewski, Pierre-Alain Chaumeil, and Philip Hugenholtz. 2018. A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nature Biotechnology* 36, 10 (2018), 996–1004.

[21] Edoardo Pasolli, Duy Tin Truong, Faizan Malik, Levi Waldron, and Nicola Segata. 2016. Machine learning meta-analysis of large metagenomic datasets: tools and biological insights. *PLoS Computational Biology* 12, 7 (2016), e1004977.

[22] Fabian Pedregosa et al. 2021. memory-profiler. https://pypi.org/project/memory-profiler/.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[24] Owen Queen and Scott J. Emrich. 2021. LASSO-based feature selection for improved microbial and microbiome classification. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2301–2308.

[25] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. 4–11.

[26] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical Bayesian optimization of machine learning algorithms. *Advances in neural information processing systems* 25 (2012).

[27] Shinichi Sunagawa, Silvia G Acinas, Peer Bork, Chris Bowler, Damien Eveillard, Gabriel Gorsky, Lionel Guidi, Daniele Iudicone, Eric Karsenti, Fabien Lombard, et al. 2020. Tara Oceans: towards global ocean ecosystems biology. *Nature Reviews Microbiology* 18, 8 (2020), 428–445.

[28] Laurens Van Der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *The journal of machine learning research* 15, 1 (2014), 3221–3245.

[29] Wei Wang, Yan Huang, Yizhou Wang, and Liang Wang. 2014. Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 490–497.

[30] Yasi Wang, Hongxun Yao, and Sicheng Zhao. 2016. Auto-encoder based dimensionality reduction. *Neurocomputing* 184 (2016), 232–242.

[31] Lei Zhu, Jaishree Tripathi, Frances Maureen Rocamora, Olivo Miotto, Rob van der Pluijm, Till S. Voss, Sachel Mok, Dominic P. Kwiatkowski, François Nosten, Nicholas P. J. Day, Nicholas J. White, Arjen M. Dondorp, Zbynek Bozdech, Aung Pyae Phyo, Elizabeth A. Ashley, Frank Smithuis, Khin Lin, Kyaw Myo Tun, M. Abul Faiz, Mayfong Mayxay, Mehul Dhorda, Nguyen Thanh Thuy-Nhien, Paul N. Newton, Sasithon Pukrittayakamee, Tin M. Hlaing, Tran Tinh Hien, Ye Htut, and Tracking Resistance to Artemisinin Collaboration I. 2018. The origins of malaria artemisinin resistance defined by a genetic and transcriptomic background. *Nature Communications* 9, 1 (2018), 5158.