Analysis of biome compositional data and BMI prediction

Siyuan Pang (676221584, spang8)

December 11th, 2020

A Report Presented for STATS 542: Statistical Learning

Final Project

University of Illinois at Urbana Champaign

1. PROJECT DESCRIPTION AND SUMMARY

The provided dataset contains demographic features and compositional features. The compositional data are from 16S ribosomal RNA (rRNA) gene and have already been normalized to appropriately scaled ratios such that the sum of all columns of each sample equals one. In this project, we are going to address sparsity issues of data first and then perform unsupervised and supervised learning methods that covered in STAT 542 classes to find potential patterns of compositional data and predict body mass index (BMI) and alcohol frequency.

The methods of this project contain unsupervised learning and supervised learning. In unsupervised learning, 3 different clustering approaches (PCA, k-means, and hierarchical clustering) are performed to understand data and identify underlying clusters and potential patterns of compositional variables. In supervised learning, 3 different classification methods (SVM, random forest, and KNN) and 2 different regression models (Lasso and random forest) are conducted to predict BMI and alcohol frequency based on either compositional data or demographic data. Parameters are tuned for each model considering bias-variance trade-off. Based on our analysis, compositional features are reasonably reduced. Correlations between features and between samples are discovered. Important compositional features as well as demographic features that are associated with BMI are targeted. The built models are able to predict BMI and alcohol frequency with low training error but relative high testing error.

2. LITERATURE REVIEW

American Gut is an open platform that collects human microbiome data from all over the world starts from 2012 to discover the kinds of microbes and microbiomes via a self-selected cohort. McDonald et al. [1] utilized the sample contribution from this platform to study human gut microbiome specimens in different countries. According to the utility of living data resources and cross-cohort comparison, they associated microbiome data with psychiatric illness to provide guidance for microbiome change during surgery. In addition, they discovered connections between race, sex, smoking history, education with

body mass index (BMI) by classification and regression models. Similarly, Li [2] also used this data to investigate the association between gut microbiome composition with long-term diet to provide suggestions for fat intake.

The microbiome data is from the 16S ribosomal RNA (rRNA) gene which is commonly used to study bacterial composition [3]. Sequence reads have been clustered into operational taxonomic units (OTUs). Each OTUs can be assigned a taxonomic lineage by comparing it with a known bacterial 16S rRNA database [2]. However, most OTUs are absent from a large number of samples. That leads to the sparsity issue in data analysis: only a small number of samples are found having organisms, while others are not detected due to insufficient sequencing depth [3]. Sparsity may lead to significant bias when people build models with scaled data. Thus, processing the high-dimensional sparse compositional data is required before data analysis.

MetagenomeSeq Bioconductor package is available in R to process the sparse OTUs data. It is designed to determine features that are differentially abundant between groups of samples. MetagenomeSeq is able to address the effects of normalization and under-sampling of microbiome data. Several studies used this method to process the high-dimensional sparse biome data [3, 4, 5]. Paulson [5] used MetagenomeSeq to perform an optimization routine that estimated probabilities that a zero value in a feature is a technical zero or not. Thus, compositional features can be reduced based on the average number of effective samples in all features. Zero replacement is performed by R zCompositions package [6]. A study [3] shows the sparsity was largely reduced while the depth of coverage was maintained by using the MetagenomeSeq method.

3. UNSUPERVISED LEARNING

3.1 Feature Engineering and Data Cleaning

The goal of unsupervised learning is to understand the data and find potential clusters. Only the OTUs variables are used as covariates in clustering.

Compositional data are very sparse. The sparsity level = 99.4% according to zero values in all covariate matrix. The 'metagenomeSeq' package is used to process the sparsity issues. Data is trimmed to reduce dimension according to abundance. As shown in Figure 1, features that less than the average number of effective samples in all features are removed, such that the features leftover present more than 1% of the subjects. The compositional data dimension becomes to 9511 x 1824. Normalization is performed due to varying depths of coverage across samples.

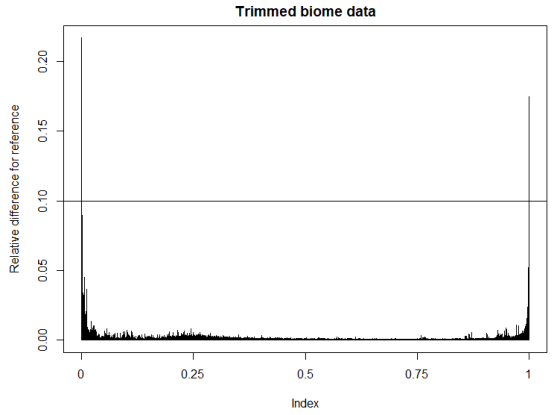


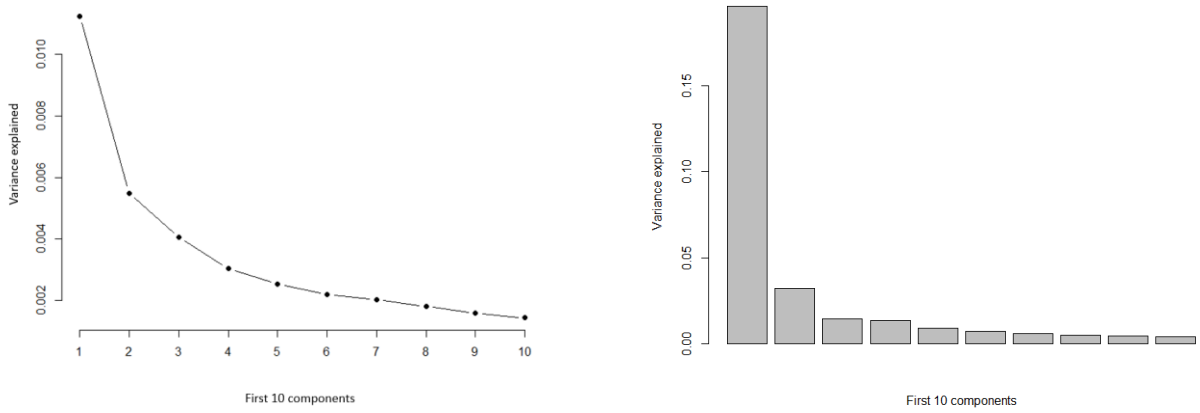
Figure 1. Features are trimmed according to abundance

After reducing the dimension, zeros in variables are replaced by cmultRepl function with ‘CZM’ method [6]. With zero values are filled, samples that are less than 1e-4% abundant in any sample are removed, resulting in the compositional variable dimension becomes 6963x1824. Lastly, compositional variables are converted to centered log-ratio ($\log(x) - \frac{1}{p}\sum_{i=1}^{p} \log(x)$) to avoid scaling problems.

Column names are cleaned in order to better understand cluster groups. By cleaning the column names, 12 unique bacteria taxa are found: "Cyanobacteria", "Bacteroidetes", "Firmicutes", "Verrucomicrobia", "Synergistetes", "Proteobacteria", "Euryarchaeota", "Actinobacteria", "Lentisphaerae", "Fusobacteria", "Tenericutes", and "Thermi".

3.2 Principle Component Analysis

Principle component analysis (PCA) is performed to find the potential relation between compositional data. Correlation is set to be true when performing PCA. Since data has already be scaled, scaling is not applied again in PCA. Both untrimmed data and trimmed data according to abundance are performed by PCA to compare how the feature engineering can affect the PCA results. Figure 2 (a) shows the variance on the first 10 principle components from all features, while Figure 2 (b) shows that from trimmed features. Though PC1 contains most of information in the two figures, only a little (1.1%) variance are projected on PC1 if PCA is performed on all features. On the other hand, in trimmed data, the first 2 components explain 19.6% and 3.2% of the variance in the data. Thus, it makes more sense to explain the data after feature engineering.



(a) (b)

Figure 2. Variance explained on the first 10 components based on (a) PCA performed on all features; (b) PCA performed on trimmed features.

The covariance biplot made from the abundance trimmed dataset are shown in Figure 3. The ray in this figure show the amount of variance exhibit by each taxon relative to the PC center. The longer rays mean more variance across samples. Among all the features, "k__Bacteria|p__Firmicutes|c__Clostridia| o__ Clostridiales|f__Ruminococcaceae|g__Ruminococcaceae|s__Ruminococcaceae-unspecified.73" has the longest ray, indicating it exhibits the most variation relative to all taxa across samples. On the other hand, "k__Bacteria|p__Firmicutes|c__Clostridia|o__Clostridiales|f__Clostridiales|g__Clostridiales|s__Clostridia

les-unspecified.215" has the shortest ray, showing this bacteria has the least variation. The correlation of the abundance of two taxa can be seen from the angle of rays. If the two rays are orthogonal, that means the 2 taxa are uncorrelated. In addition to discover the relation between features, we can also produce the biplot to find relation between samples by performing PCA on transposed matrix. The results show sample 2355 is the least similar to any other sample because it is furthest from the PC center. The sample 654 is the most similar to any other sample as it is the closest to the center.

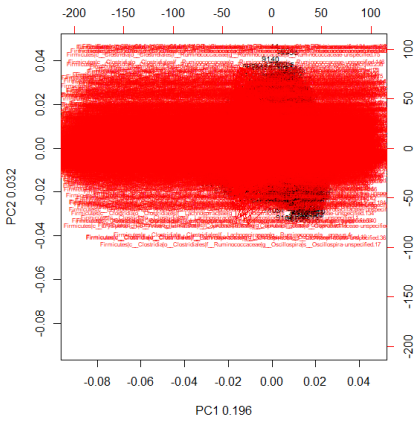


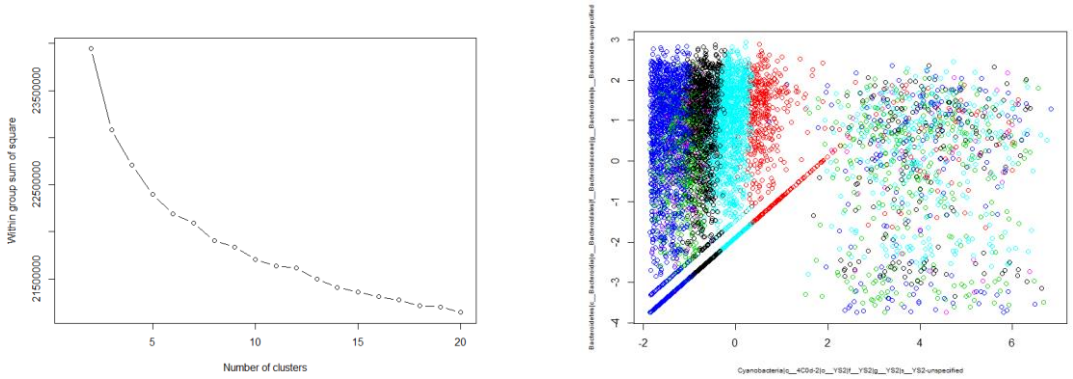
Figure 3. PCA covariance biplot of the trimmed abundance data

3.3 K-means Clustering

K-means clustering is performed to find the potential number of clusters from compositional data. As k-means method is sensitive to the initialization of cluster centers, the number of clusters should be defined first. The sum of square distances within clusters (WSS) are calculated by varying the number of clusters from 2 to 20. According to the Figure 4 (a), the WSS decreases as the number of clusters increases. The drops are large until we have 6 clusters. Thus, a k-means model is built with the number of clusters = 6. Figure 4 (b) is an example of how two variables are clustered. As the variables are ordered by the abundance, the first 2 variables are plot showing how six clusters are defined.

It is worth noticing that the number of clusters is still hard to determine. Other studies such as Yan et al. defined 7 clusters from 20214 RNA sequence and Patel et al defined 5 clusters from 5948 RNA sequence

[7]. As k-means results highly depend on the initiation of the number of clusters, this method may not well present the clustering results.



(a) (b)

Figure 4. (a) Sum of square distance within clusters change with number of clusters in k-means; (b) The most two abundant variables are clustered into six groups.

3.4 Hierarchical Clustering

Hierarchical clustering is a popular method to show correlations between variables. Figure 5 is the hierarchical cluster dendrogram. Euclidian distance and the ward.D2 method [8] is used to cluster groups together by their squared distance from the geometric mean distance of the group. According to the k-means clustering results that compositional data are divided into six groups, colors in Figure 5 shows the hierarchical clustering for six different groups.

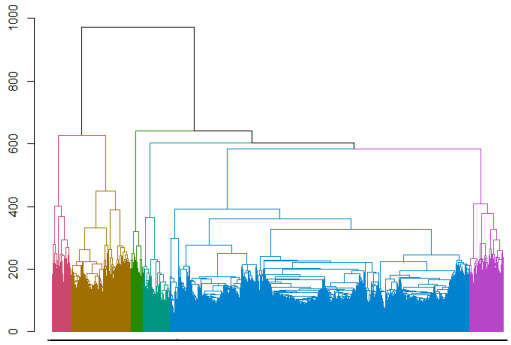
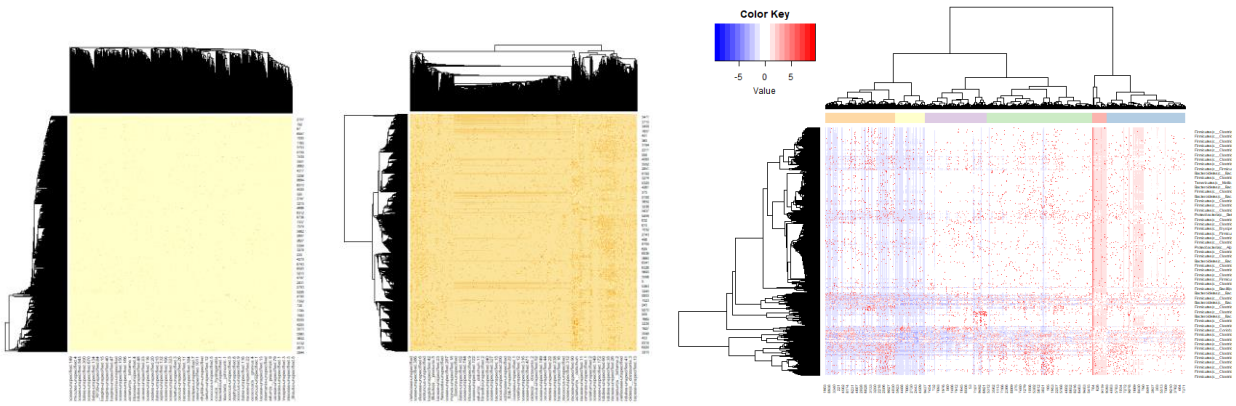


Figure 5. Hierarchical cluster dendrogram

The heatmaps are plot in Figure 6 helps discover potential patterns. It recognizes the samples and taxa based on the clusters. The darker regions imply those data are more correlated. We can take those as subgroups of data. Observing the results calculated from data before addressing sparsity, little relations or patterns can be seen from the heatmap as shown in Figure6 (a). On the other hand, more correlation present after solving the sparsity problems. Thus, the sparsity issue largely affects clustering results. To better illustrate the results, Figure 7 is a new plot heatmap with more clear indication. White values indicate no relation. Row color labels are plot with six colors indicate OTUs taxonomic classes. There are still a lot of regions that are white. Further dimension reduction and subgroup sample analysis could be carried out to reduce computation time.



(a) (b) (c)

Figure 6. Heatmap showing correlation between features and between samples for (a) data before process sparsity issues; (b) data after processing sparsity issues; (c) data after processing sparsity issues and plot in six clusters

4. SUPERVISED LEARNING

4.1 Classification – SVM

In this section, support vector machine (SVM) is used to classify BMI categorical data with compositional variables as inputs. Non-linear SVM with kernel trick is chosen because our data have high-dimensional features. The kernel function to be used is the radial basis function (RBF) kernel. Based

on the unsupervised learning analysis, we learned that we can further reduce the dimension and subgroup samples to build classification models. MetabiomeSeq method is used again to shrink the feature size. Imputation is performed and data are converted to centered log-ratio. The missing rate in bmi_cat = 5.7%. Thus, those amounts of observations are removed.

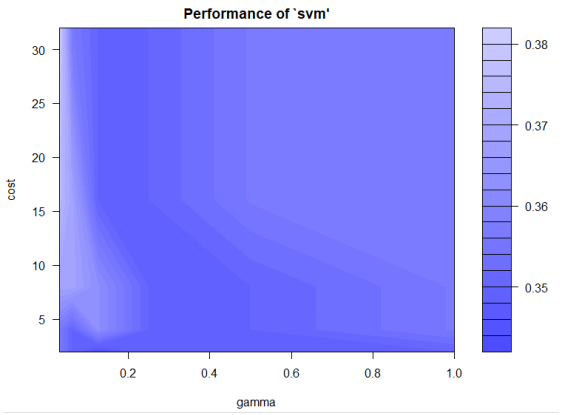


Figure 7. Optimize the tuning parameter γ and cost in SVM

There are two tuning parameters in the SVM model: γ and cost. A grid of parameters γ and cost are tuned by 10-fold cross validation to get the best performance for the model. The best performance as shown in Figure 7 is found when γ = 0.125 and cost = 2. Table 1 shows the confusion matrix of SVM classification results. The fitted SVM model with RBF kernel gives the prediction accuracy = 79.5% with a confidence interval between 76% - 83%. However, the testing error only reaches 47%, which is similar to the study [9] who got SVM accuracy 43.1%. To improve the testing accuracy, careful inspection for feature and subgroup sample selection is required due to sample noise and complexity.

Table 1. Confusion matrix of SVM classification

	Reference			
Prediction	Normal	Obese	Overweight	Underweight
Normal	270	15	50	18
Obese	4	34	0	0
Overweight	10	0	53	0
Underweight	1	0	0	23

4.2 Classification – Random Forest

According to Harris's [9] results, random forest gave better predicting accuracy than SVM. Thus, random forest is performed to predict alcohol frequency with compositional data as inputs. The compositional data processing is the same as that is performed in SVM section. The missing rate in alcohol frequency = 1.38%. Those missing observations are deleted.

Three parameters are tuned according to 10-fold cross-validation.: 1) mtry, represents the number of variables tried at each split of the model; 2) ntree, increases the accuracy up to some point; 3) nodesize, indicating the minimum node size and the depth of trees. The optimal parameters are set as mtry = 3, ntree = 200, nodesize = 3. The classification results are shown in Table 2. The random forest model has accuracy 96.8% with 0.95 confidence interval between 94% - 98%, which is better than the SVM results.

Table 2. Confusion matrix of random forest classification

	Reference				
Prediction	Daily	Never	Occasionally (1-2 times/week)	Rarely (a few times/month)	Regularly (3-5 times/week)
Daily	46	0	0	0	0
Never	0	113	0	0	0
Occasionally (1-2 times/week)	4	3	126	5	4
Rarely (a few times/month)	0	0	0	109	0
Regularly (3-5 times/week)	0	0	0	0	90

4.3 Classification – KNN

K-nearest-neighbor (KNN) is performed to predict bmi_cat using demographic variables race and sex as inputs. The missing values in *X* and *y* are removed. The parameter k is tuned with a grid 1 to 30. The testing error of the KNN model is 57.4% as shown in Table 3. KNN is not a good choice in this situation. Inputs race and sex are categorical data. They are transformed into dummies in prediction. While the output is also categorical data, the model results have too many ties in probability prediction. Thus, almost all the classification become "normal".

Table 3. Confusion matrix of KNN classification

	Reference			
Prediction	Normal	Obese	Overweight	Underweight
Normal	956	160	401	144
Obese	0	0	0	0
Overweight	7	3	10	3
Underweight	0	0	0	0

4.4 Regression – Lasso

Lasso regression is performed to predict numeric BMI values with compositional variables as inputs. Since our data is sparse with high-dimensional features, Lasso is able to shrink some of the coefficients to 0 if the effect of that variable is small, makes the computation more efficient. Also, by tuning the parameter λ, bias-variance trade-off can be adjusted to prevent overfitting.

Missing values that occupy 2.6% in BMI are removed. In addition, there are obvious invalid BMI values which are extremely large. BMI that are larger than 100 are removed. 10-fold cross validation is performed to find the optimal λ. Figure 8 shows how error change with log λ and the optimal $\lambda = 0.8485$. The absolute values of β associated with the most important features are shown in Figure 9. Since the name of compositional data are too long to exhibit in *x* labels, the *x* label feature names are presented at the right-hand side in order. We can conclude that "Firmicutes.c__Clostridia.o__Clostridiales.f__ Clostridiales.g__Clostridiales.s__Clostridiales.unspecified.11", "Firmicutes.c__Clostridia.o__ Clostridiales.f__Lachnospiraceae.g__Dorea.s__Dorea.unspecified.3", and "Firmicutes.c__Clostridia.o __Clostridiales.f__Clostridiales.g__Clostridiales.s__Clostridiales.unspecified.34" are the three most important features in determining the BMI value in Lasso model. The MSE of training data is 39.31, and that if testing data is 46.64.

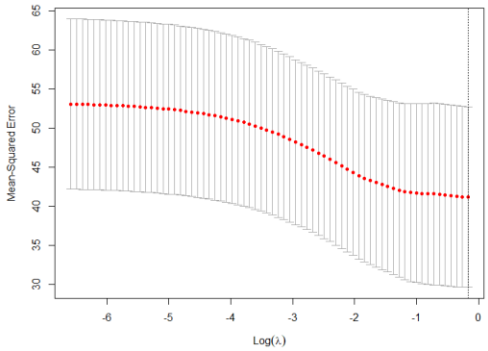


Figure 8. Determine tuning parameter λ in Lasso regression

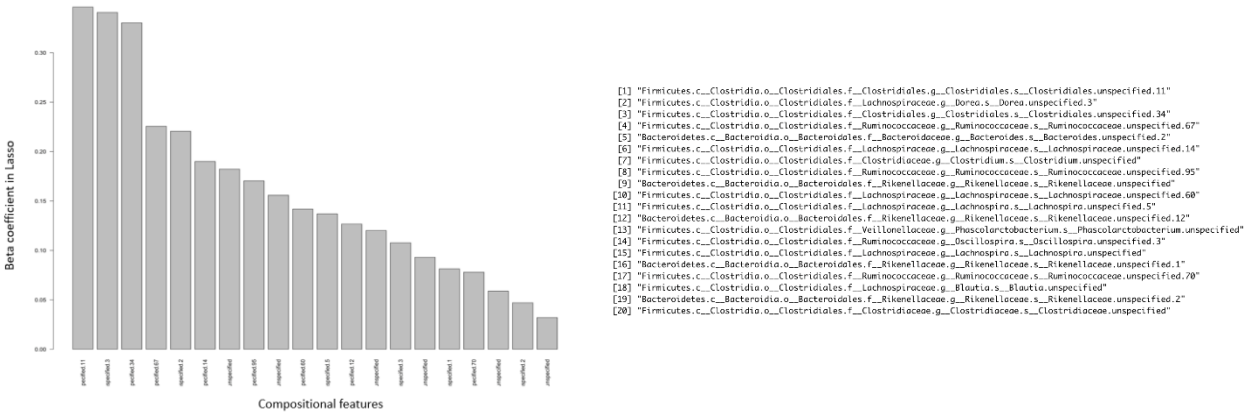


Figure 9. Important features in Lasso regression model

4.5 Regression – Random Forest

Random forest is implemented to predict BMI with age, weight, race, and sex as inputs. Data cleaning is performed: 1) Observations that contain "Not provided" are removed. 2) In the age column, 'child' is defined as 10s, and 'teen' is defined as 20s. Then ages are transformed into numeric type. 3) In the weight column, the weight with value 0 is removed. Also, weight is transformed into numeric type. 4) Race and sex are transformed into factors. 5) In the BMI column, remove invalid values that are larger than 100, and turn the data type into numeric.

Parameters are tuned for the number of trees (ntree), selective features for splitting trees (mtry), and splitting criteria nodesize. By calculating a grid of these 3 parameters, and comparing the mean squared

error, the optimal tunning parameters are: ntree = 1000, mtry = 2, and nodesize = 10. A random forest model is then built based on the optimal parameters. Feature importance can be seen in Figure 10. Weight is the most important feature that determines BMI, while race is the least important feature. The MSE of training dataset = 7.05, and the MSE of testing dataset = 9.47.

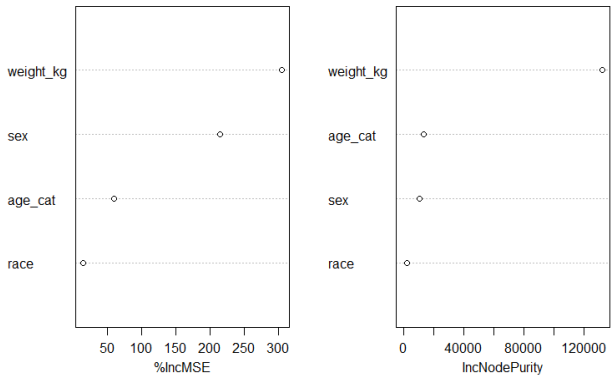


Figure 10. Important features in random forest model

5. COLLABORATOR'S QUESTION

In this project, we are targeting the 16S rRNA compositional data as variables to find their potential clusters or patterns. However, the OTUs data are very sparse which have little signals, may lead to significant clustering or prediction bias. To solve the sparsity issues, MetagenomeSeq Bioconductor package in R is performed to address zero values in datasets. compositional variables that less than the average number of effective samples in all variables are removed, such that the features leftover present more than 1% of the subjects. Then the remaining zero values are imputed by CZM method such that sparsity = 0%. Lastly, compositional variables are converted to centered log-ratio to avoid scaling problems. With this signal processing, sparsity issues can be solved, and findings are more reliable based on the heatmap comparison in Figure 6 (a) and (b).

The findings in unsupervised learning are hard to be justified as there's no exact truth to verify results. Clustering results are sensitive to the number of clusters predefine in the model. To make the clustering approaches more convincing, the loss vs. clusters are plot in Figure 4 (a). Clustering algorithms are run by

increasing the number of clusters and calculate within group sum of distance. The distance decreases accordingly. The number of clusters can be approximately determined when the slope of decreasing distance becomes small.

Several classification and regression models are built to predict BMI, BMI_cat, and alcohol frequency. Cross-validation or hyperparameter tuning are commonly used to find the optimal parameters in supervised learning models. Different models are compared to achieve less prediction errors. However, results are always biased after we try different tuning parameters and different models. Thus, in addition to the above two approaches to reduce bias, we can consider 1) Feed more data into models; 2) Try different methods to treat missing values and outliers; 3) Try different feature selection methods. In this work, MetagenomeSeq package is used to select features for compositional data. There are many other methods can perform feature selection for high-dimensional data. Further studies could be conducted to compare how different feature selection methods affect prediction results.

6. REFERENCE

[1] McDonald, Daniel, et al. "American Gut: an open platform for citizen science microbiome research." *Msystems* 3.3 (2018): e00031-18.

[2] Li, Hongzhe. "Microbiome, metagenomics, and high-dimensional compositional data analysis." *Annual* Review of Statistics and Its Application 2 (2015): 73-94.

[3] Paulson, Joseph N., et al. "Differential abundance analysis for microbial marker-gene surveys." *Nature methods* 10.12 (2013): 1200-1202.

[4] Paulson, Joseph Nathaniel, Mihai Pop, and Hector Corrada Bravo. "metagenomeSeq: Statistical analysis for sparse high-throughput sequencing." *Bioconductor package* 1.0 (2013): 191.

[5] Wagner, Justin, et al. "Interactive exploratory data analysis of Integrative Human Microbiome Project data using Metaviz." *F1000Research* 9.601 (2020): 601.

[6] Palarea-Albaladejo, Javier, 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 (2015): 85-96.

[7] Qi, Ren, et al. "A spectral clustering with self-weighted multiple kernel learning method for single-cell RNA-seq data." *Briefings in Bioinformatics* (2020).

[8] Clooney, Adam G., et al. "Comparing apples and oranges?: next generation sequencing and its impact on microbiome analysis." *PloS one* 11.2 (2016): e0148028.

[9] Harris, Zachary N., et al. "Massive metagenomic data analysis using abundance-based machine learning." *Biology direct* 14.1 (2019): 12.