Multivariate statistical analysis of metagenomic data

Boris Shilov

Introduction

The aim of this analysis is to explore the functional space of the metagenomic dataset assembled by Forslund et al. (n.d.) - this is a collection of measurements of the gut microbiota of different patients, primarily profiled for whether they contain particular genetic sequences and identified taxonomic groups. My primary objective here is to determine, using dimensionality reduction and ordination methods, whether there is some structure in the data, and to see whether this structure is in any way associated with the treatment/disease status of the samples, and to do this without taking taxonomic information into account (hence the limitation to functional space).

Data format

```
library(cowplot)
library(tidyverse)
set.seed(100)
data = read_tsv("data.r")
```

The dataset consists of six columns. The first is sample IDs, which are not unique keys since every sample may have multiple features. The second is the dataset of origin, which is either Danish (MHD), Swedish (SWE) or Han Chinese (CHN). The third column is treatment status of a particular patient, where they are either healthy (ND CTRL), are having type 2 diabetes treated with metformin included (T2D metformin+), are having type 2 diabetes treated without metformin (T2D metformin-) or are having type 1 diabetes treated (T1D). The fourth column specifies the type of feature in the fifth column. This is either gut microbial/metabolic module (GMM), SEED database annotation, bacterial family, bacterial genus or metagenomic operational taxonomic units (Motu). The fifth column contains the feature names which are coded differently depending on the feature type. Finally, the sixth column contains the abundance of a given feature.

The feature types are split into two spaces: the taxonomic space represented by the genuses, families and taxonomic units, and the functional space as annotated by SEED and GMM. The full SEED and GMM feature code annotations can be found in the appendix files, taken from Supplementary Table 10 of Forslund et al. (n.d.). We are only interested in exploring the structure of the functional space.

```
functional_space = filter(data, FeatureType %in% c("GMM", "SEED"))
functional_space
```

```
## # A tibble: 413,168 x 6
##    Sample Dataset Status  FeatureType Feature Abundance
##    <chr>  <chr>   <chr>   <chr>       <chr>       <dbl>
##  1 MH0443 MHD     ND CTRL GMM         MF0119        902
##  2 MH0443 MHD     ND CTRL GMM         MF0090        302
##  3 MH0443 MHD     ND CTRL GMM         MF0097          0
##  4 MH0443 MHD     ND CTRL GMM         MF0092          0
##  5 MH0443 MHD     ND CTRL GMM         MF0011       1076
##  6 MH0443 MHD     ND CTRL GMM         MF0083          3
##  7 MH0443 MHD     ND CTRL GMM         MF0110          0
##  8 MH0443 MHD     ND CTRL GMM         MF0063          4
##  9 MH0443 MHD     ND CTRL GMM         MF0042          1
## 10 MH0443 MHD     ND CTRL GMM         MF0024        185
## # ... with 413,158 more rows
```

In order to actually perform analysis, we transform the data into the data matrix X, with every row being a sample and every column a particular feature. First we combine redundant columns. Some ecological statistical functions will assume the transpose of this format, so we will stick to using a data frame after our initial processing here - tibbles are not transposed easily due to not implementng row names.

```
united_functional_space = functional_space %>%
  unite(SampleInfo, Sample, Dataset, Status, sep="~") %>%
  unite(FeatureInfo, FeatureType, Feature, sep="~")
X = united_functional_space %>% group_by(FeatureInfo) %>% spread(FeatureInfo, Abundance)
```

There is significant missingness in 25 of the individuals, and we exclude them from further analysis.

```
X = drop_na(X)
separated_X = X %>% separate(SampleInfo, into=c("Sample", "Dataset", "Status"), sep="~")
X.df = X %>% column_to_rownames("SampleInfo") %>% as.data.frame()
```

```
sum(X==0)/(dim(X)[1] * dim(X)[2])
```

```
## [1] 0.3054907
```

30% of our matrix is sparse.

Analysis

Log transformation of the data

As we have an abudance matrix, we need to transform the data to be linear. We can do this using a log transformation, in particular the centered log ratio transform. In order to log transform, we first need to get rid of the zeroes, which we use Bayesian multiplicative replacement for. Normally we would have to normalise and filter the data for low-abundance samples, but this has already been done for us.

```
library(zCompositions)
X.df.czm = cmultRepl(X.df,  label=0, method="CZM")
```

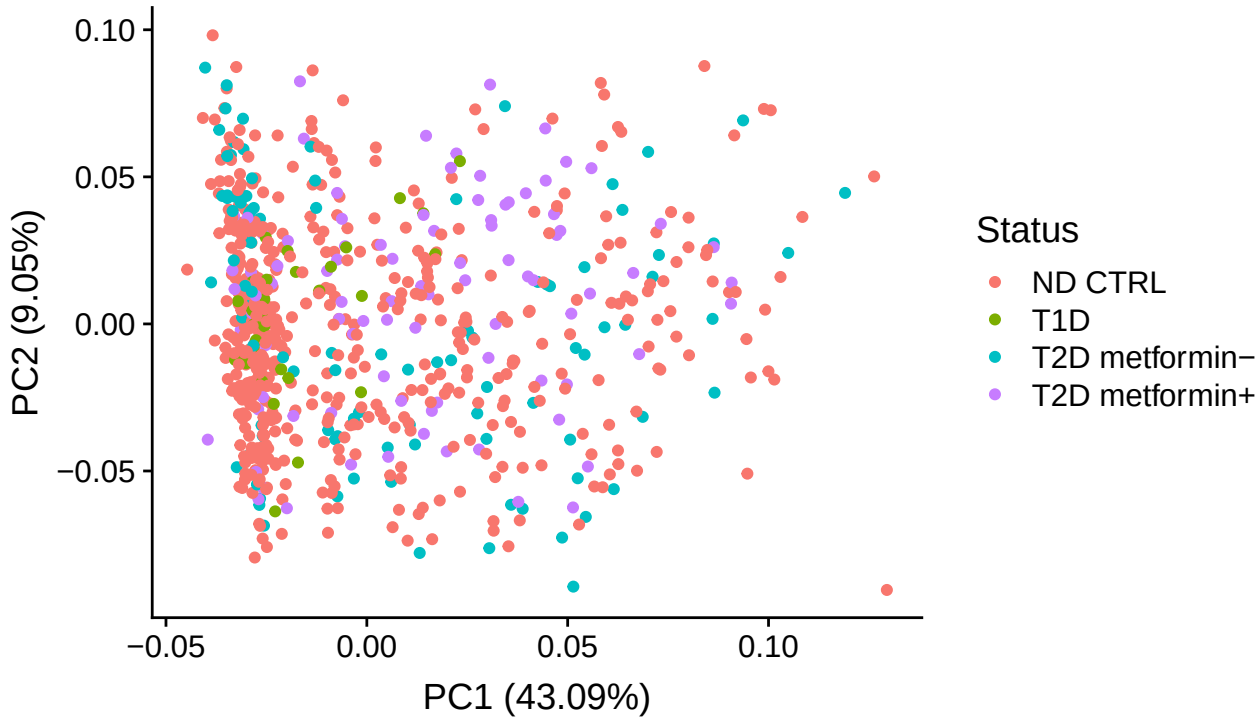
```
## No. corrected values:  29
```

```
X.df.clr = t(apply(X.df.czm, 1, function(x) {log(x) - mean(log(x))} ))
annotated_X.clr = bind_cols(separated_X[1:3], as.tibble(X.df.clr))
```

PCA

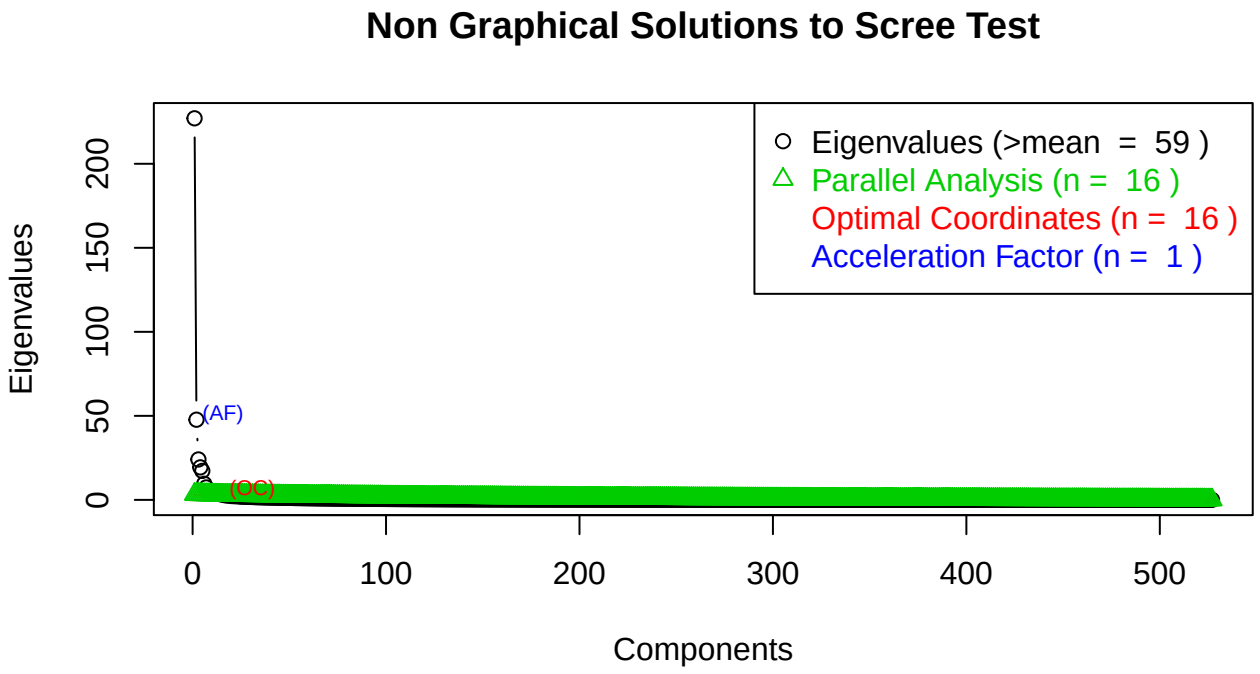
We can attempt PCA on the transformed data. We scale to unit variance.

```
library(ggfortify)
library(nFactors)
library(GGally)
X.df.clr.PCA1 = prcomp(X.df.clr, scale=T)
autoplot(X.df.clr.PCA1, data=annotated_X.clr, colour="Status", x=1, y=2)
```



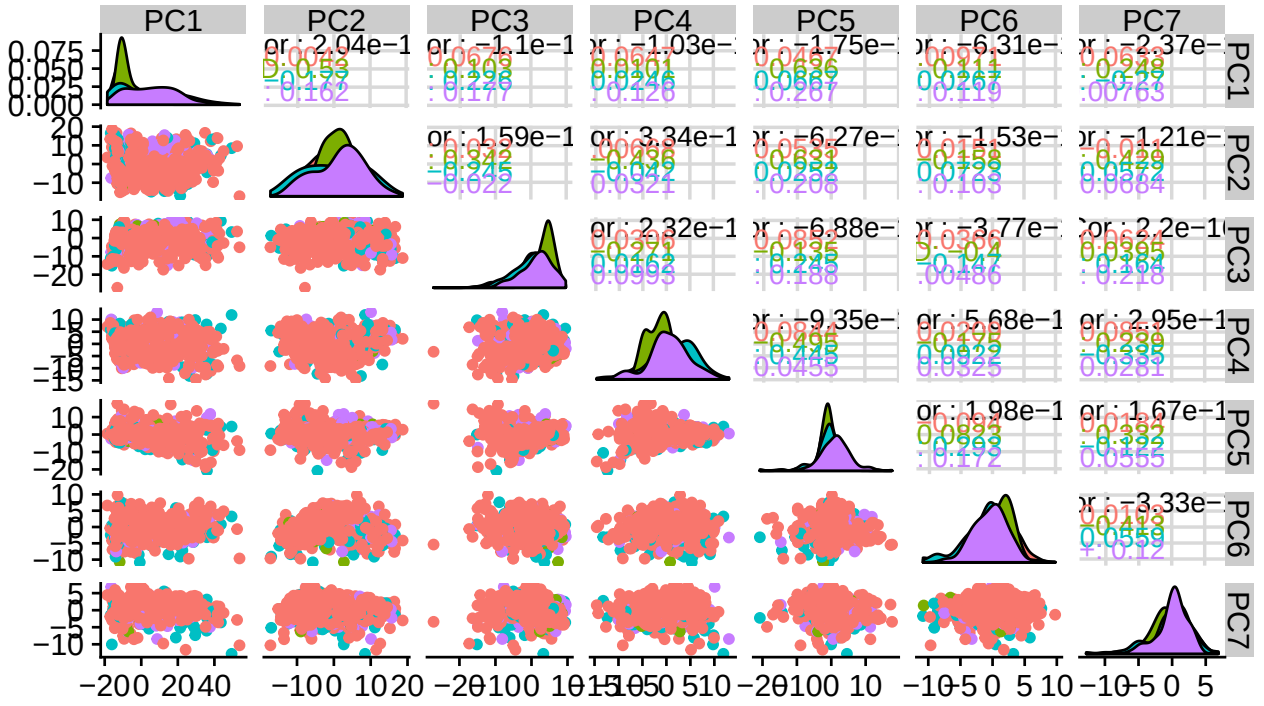
We can see that the first component accounts for a lot of variance. It would be informative to attempt parallel analysis to determine the number of principal components to retain. This technique is a formalisation of the Scree plot and works by simulating a random data frame with the same number of of samples and variables as in the original data frame, computing a correlation matrix for this random d.f. We use the eigenvalues from this simulation to decide if the components in our real analysis are likely to be random noise.

```
parallelAnal1 = parallel(subject=nrow(X.df.clr), var=ncol(X.df.clr), rep=100)
screeTest1 = nScree(x=eigen(cor(X.df.clr))$values, aparallel=parallelAnal1$eigen$qevpea)
plotnScree(screeTest1)
```

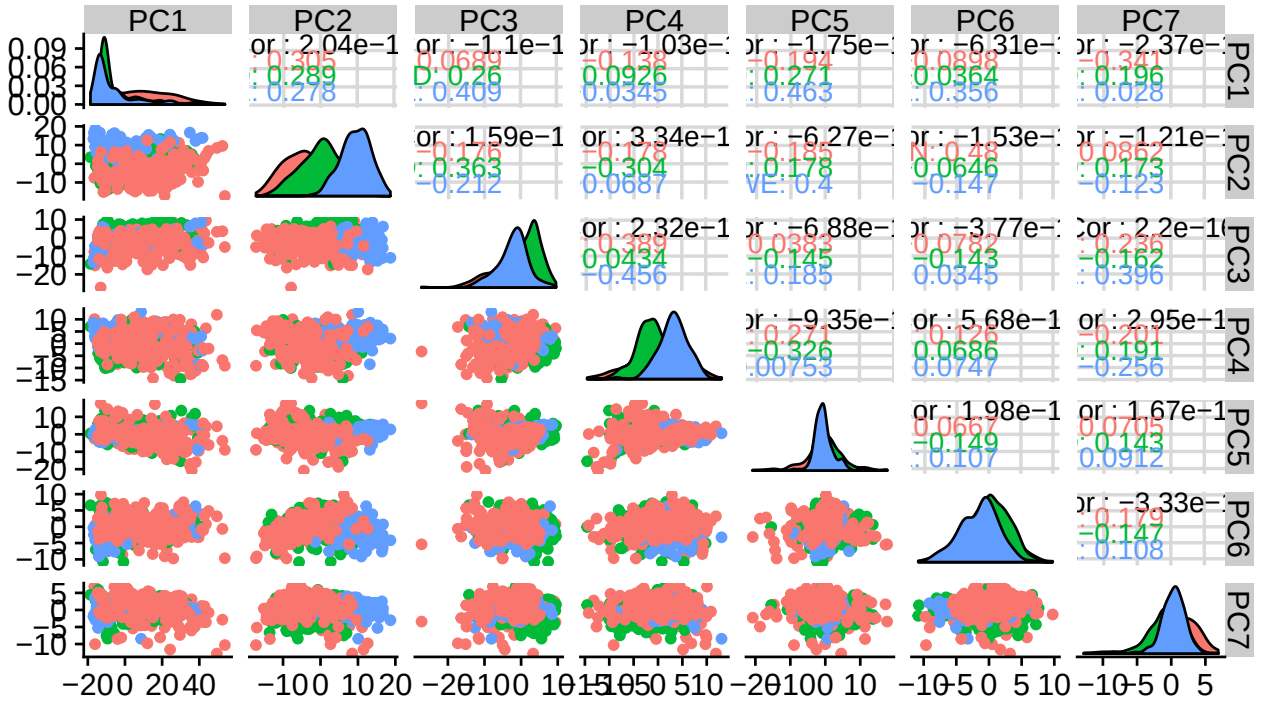


This analysis suggests around 16 components would capture most of the non-random variability. We can plot at least the first few components against each other more or less coherently.

```
PCA1_and_desc = bind_cols(as_tibble(X.df.clr.PCA1$x), separated_X[1:3])
ggpairs(PCA1_and_desc, aes(colour=Status), columns=1:7, progress = FALSE)
```




```
ggpairs(PCA1_and_desc, aes(colour=Dataset), columns=1:7, progress = FALSE)
```

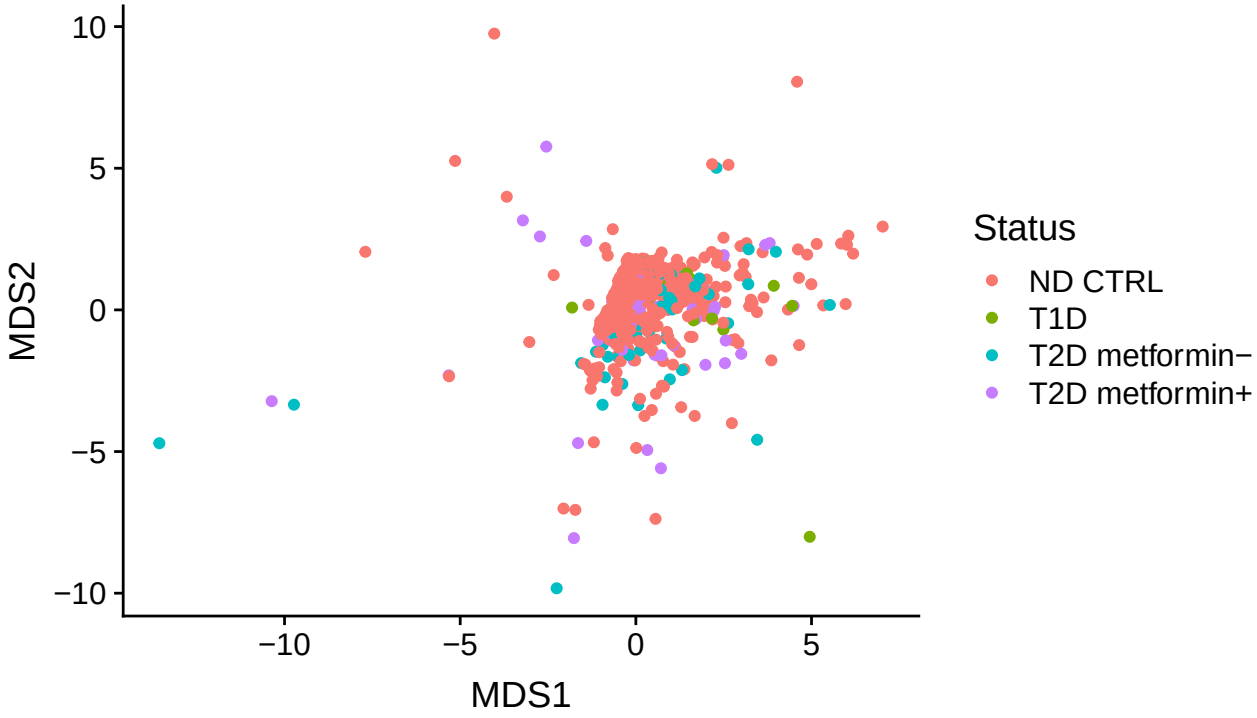


Non-Metric Multidimensional Scaling

We use the Canberra distance, a weighted Manhattan distance.

```
library(vegan)
mds1 = metaMDS(t(X.df), k=2, trymax=50, distance="canberra")
mds1_results = as.tibble(mds1$species, rownames="SampleInfo")
mds1_results_separated = mds1_results %>%
  separate(SampleInfo, into=c("Sample", "Dataset", "Status"), sep="~")
```

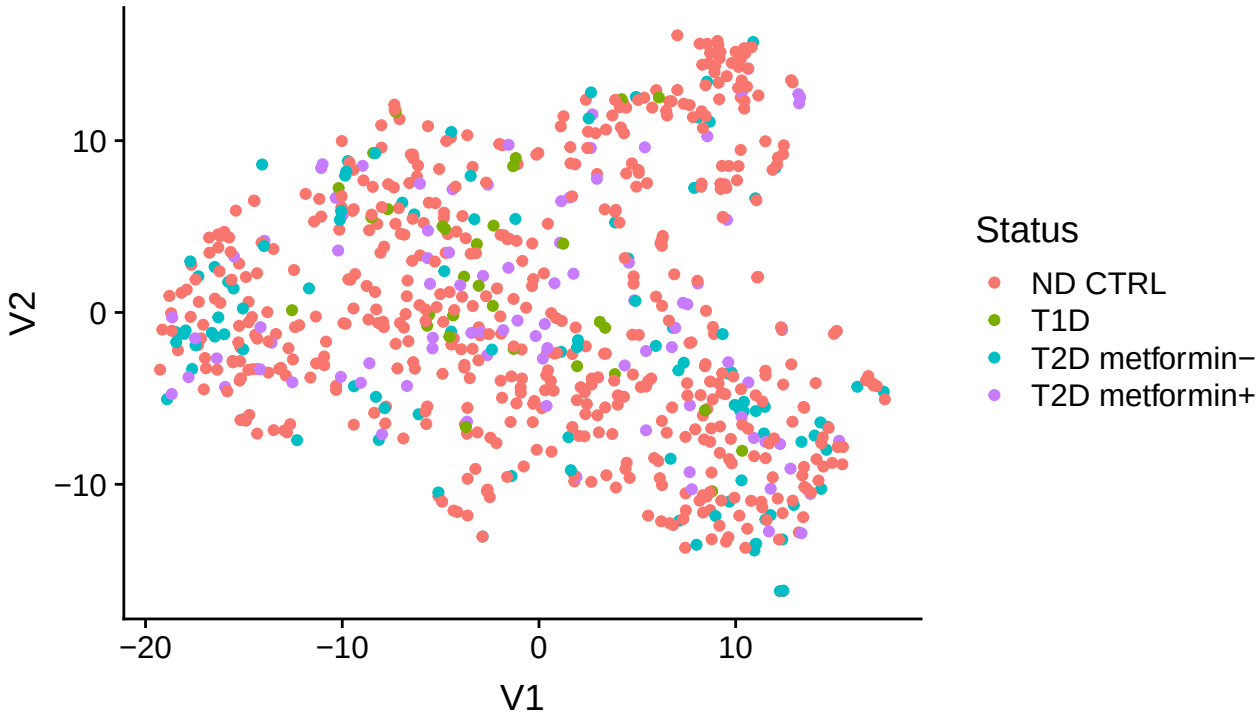
```
ggplot(mds1_results_separated, aes(x=MDS1, y=MDS2,colour=Status)) + geom_point()
```



tSNE

t-Distributed Stochastic Neighbor Embedding is a non-linear dimensionality reduction method that has demonstrated good performance in practice. We embed the data into two dimensions.

```
library(Rtsne)
tsne1 = Rtsne(X, dims=2, perplexity=30, max_iter=400)
tSNE1_res = as_tibble(tsne1$Y)
tsne_with_additionals = bind_cols(tSNE1_res, separated_X[1:3])
ggplot(tsne_with_additionals, aes(x=V1, y=V2,colour=Status)) + geom_point()
```



Conclusion

I have not managed to obtain good sample groups that correspond to clinical status. Instead, all methods attempted showcase very good grouping by dataset - this being the geographic location where the samples were collected. This "batch" effect completely drowned out the clinical associations I was interested in. The samples were taken from different age groups, ethnicities, and genders - information not available to us, that contributes to the batch effect. I made an attempt to use constrained ordination to try and condition on Dataset, but found this quite conceptually challenging and the results inconclusive, hence it is not presented in the body of this report.

In conclusion, it is clear that metagenomic data are highly complex and should not be underestimated if one wants to derive any actual signal. Such should be attempted only by experienced analysts.

Bibliography

Forslund, Kristoffer, Falk Hildebrand, Trine Nielsen, Gwen Falony, Le ChatelierEmmanuelle, Shinichi Sunagawa, Edi Prifti, et al. n.d. "Disentangling Type 2 Diabetes and Metformin Treatment Signatures in the Human Gut Microbiota." *Nature* 528: 262 EP. https://doi.org/10.1038/nature15766.