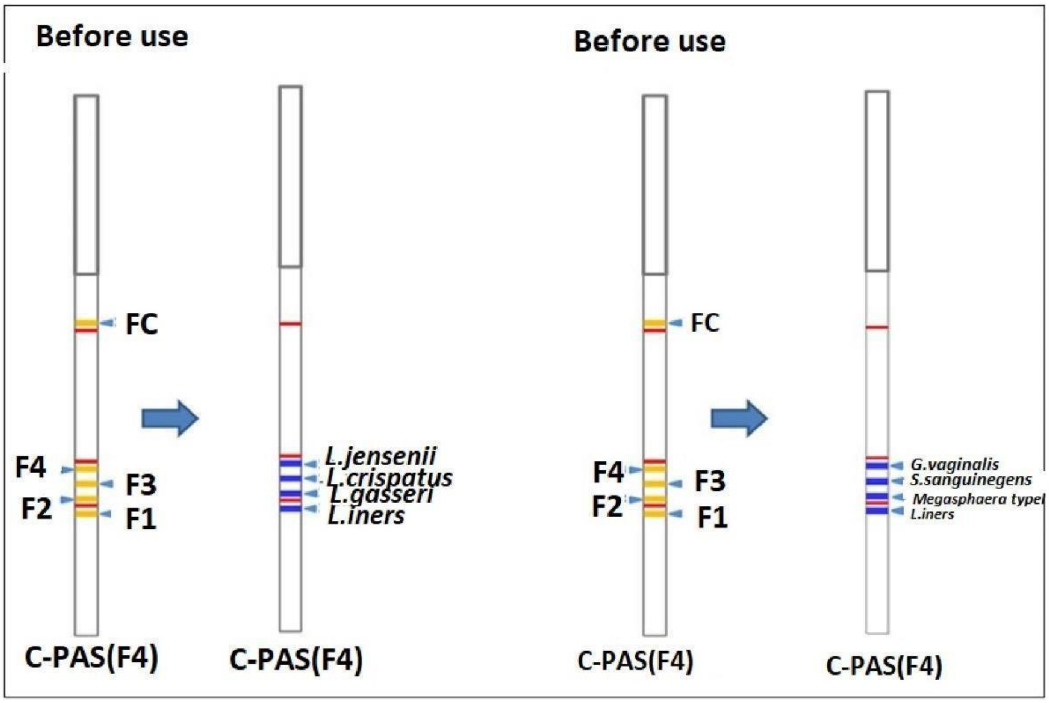
iScience, Volume *27*

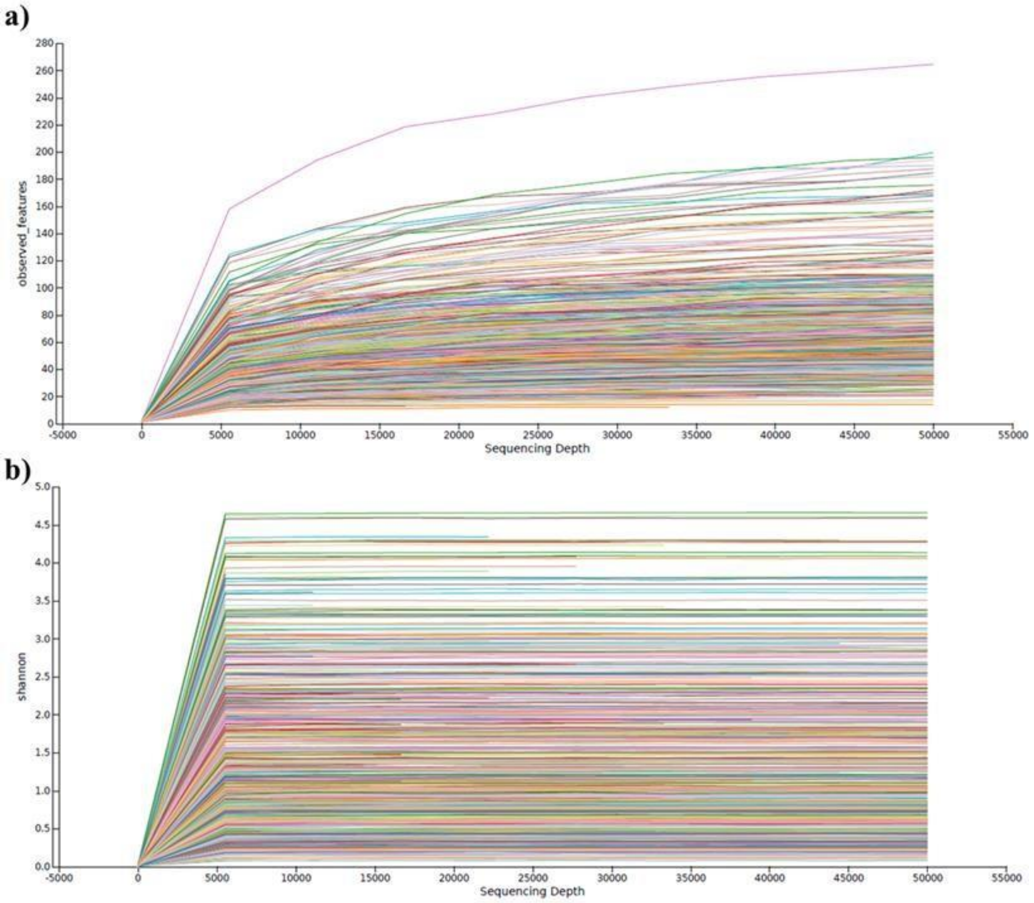
Supplemental information

Previse preterm birth in early pregnancy through vaginal microbiome signatures using metagenomics and dipstick assays

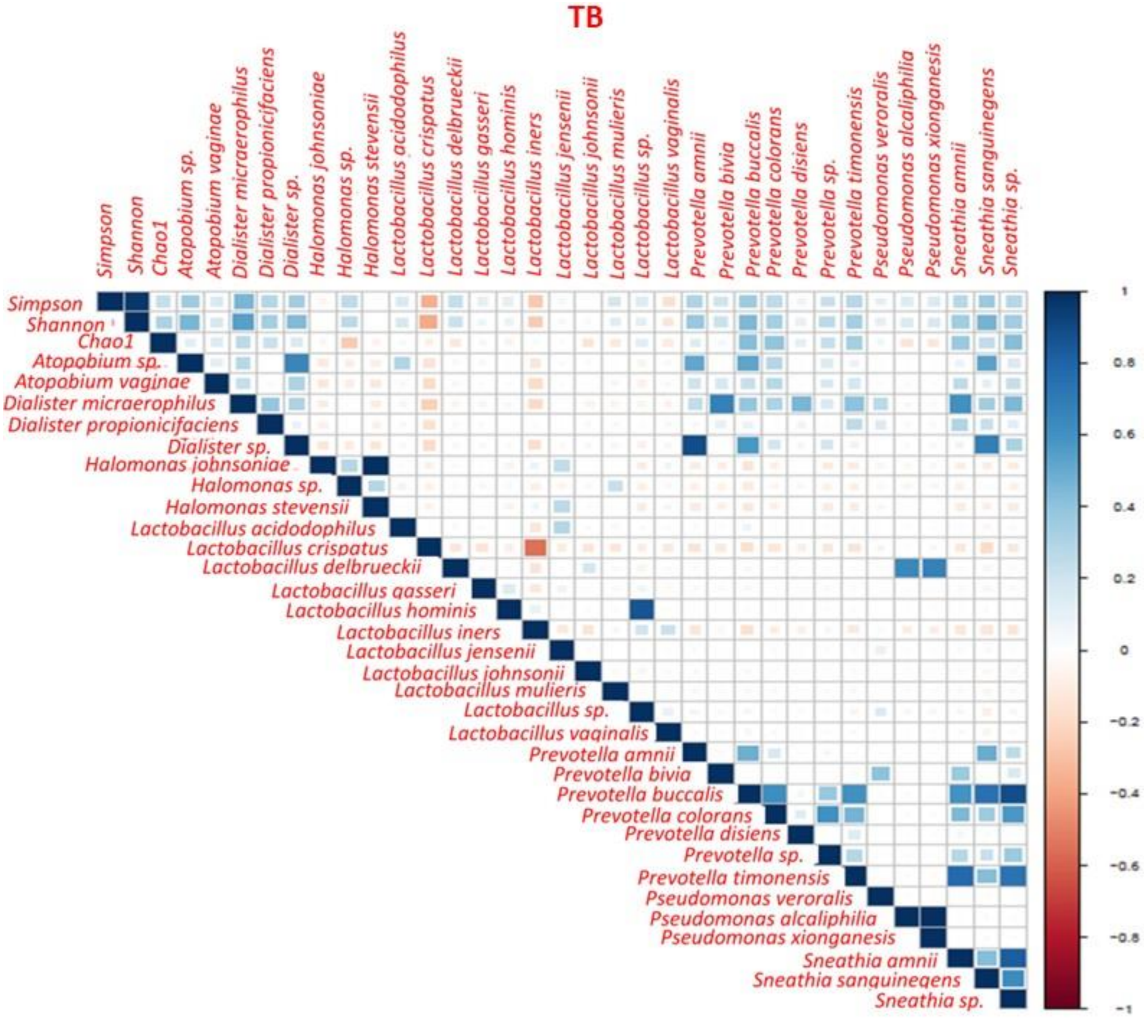
Daizee Talukdar, Mousumi Sarkar, Taruna Ahrodia, Shakti Kumar, Debjit De, Shankha Nath, Pradipta Jana, Jyoti Verma, Ojasvi Mehta, Akansha Kothidar, J.R. Yodhaanjali, Komal Sharma, Susmita Bakshi, Upma Singh, Pallavi Kshetrapal, Nitya Wadhwa, Ramachandran Thiruvengadam, Garbh-Ini study group, G. Balakrish Nair, Shinjini Bhatnagar, Souvik Mukherjee, and Bhabatosh Das



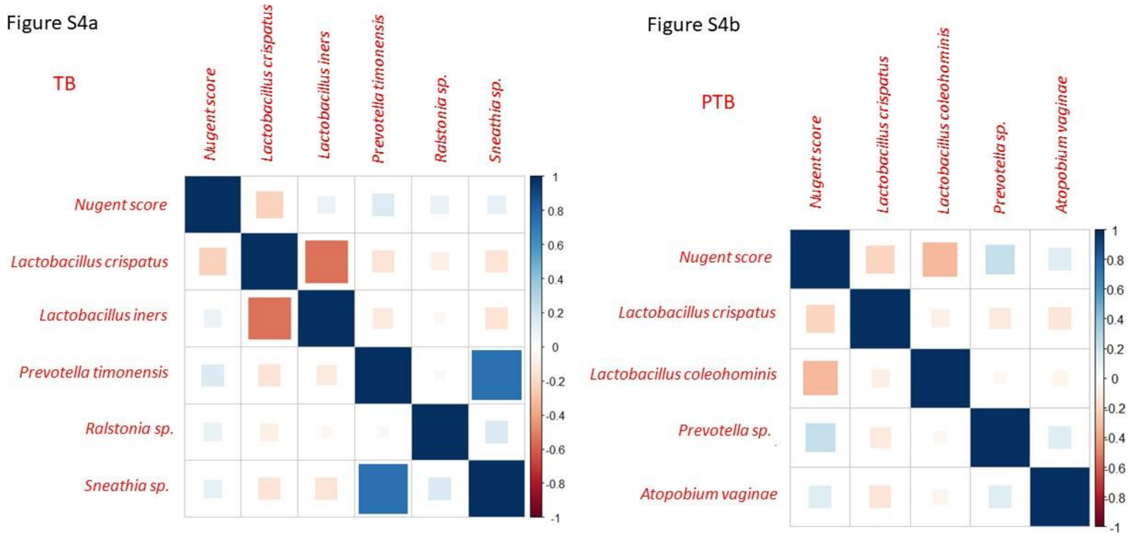
Supplementary Figure S1: **Dipstick showing the location of the probe specific to the particular bacteria, related to Figure 12**. The left strip showing *L.crispatus*, *L.iners*, *L.gasseri* and *L.jensenii* for term birth (TB) and the extreme right strip showing *G.vaginalis*, *S. sanguinegens*, *Megasphaera typeI* and *L.iners* for preterm birth (PTB).



Supplementary Figure S2: Rarefaction plot to understand the minimum number of reads/samples required for estimating intra-individual diversity within that sample i.e., the alpha diversity index reached a plateau, related to Figure 2. In our data we observed that ~ 5000reads/samples are sufficient to reach a plateaue a) Observed features b) Shannon index



Supplementary Figure S3: Correlation between vaginal microbial species and Shannon diversity index using non parametric Spearman's rank correlation test in preterm birth (TB), related to Figure 9. Blue squares represent strong positive correlation, red squares represent strong negative correlation and white squares represent non-significant correlations.



Supplementary Figure S4: Correlation between vaginal microbial species and Nugent score of vaginal samples using non parametric Spearman's rank correlation test in term samples (4a) and preterm samples (4b), related to Figure 8. Only Significant correlations were plotted here. Blue squares represent significantly positive correlation, red squares represent significantly negative correlation.

Table S1: Table showing Results of Linear Mixed Effect analysis related to Figure 10: The table contains Birth Type effect (p value), trimester effect (p value), and trimester:BirthType (p value) on species relative abundances.

Name of taxa	**BirthType effect (p value)**	**trimester effect (p value)**	**trimester:BirthType (p value)**
Atopibium vaginae	0.799	0.544	0.428
Dialister microaerophilus	0.295	0.106	0.439
Dialister propionis	0.556	0.101	0.746
gardnerella leopoldi	0.07	0.039	0.104
Gardnerella sp.	0.215	0.221	0.996
Gardnerella swidindki	0.126	0.088	0.186
Gardnerella vaginalis	0.749	0.468	0.966
halomonas johnsoniae	0.041	0.045	0.172
Halomonas stevensi	0.035	0.038	0.16
Halomonas sp.	0.094	0.17	0.16
Lactoabcillus jensenii	0.813	0.969	0.562
Lactoabcillus mulleries	0.437	0.414	0.404
Lactobacillus coleohominis	0.072	0.077	0.118
lactobacillus crispatus	0.065	0.029	0.312
Lactobacillus delbruckii	0.374	0.883	0.47
Lactobacillus hominis	0.514	0.737	0.624
Lactobacillus iners	0.214	0.497	0.747
Lactobacillus johnsonii	0.5	1	0.983
Lactobacillus sp.	0.052	0.096	0.05
Lactobacillus vaginalis	0.052	0.096	0.05
Lactobacillus mucosae	0.223	0.536	0.542
Lactobacillus hominis	0.514	0.737	0.624
Lactobascillus acidophilus	0.428	0.115	0.381
Lactobacillus gasseri	0.636	0.806	0.669
Megasphaera Ironae	0.454	0.566	0.558
prevotella amni	0.897	0.607	0.659
Prevotella buccalis	0.578	0.987	0.719
Prevotella bivia	0.277	0.821	0.326
prevotella colorans	0.063	0.041	0.129
Prevotella copri	0.526	0.6	0.894
Prevotella corporis	0.021	0.018	0.098
Prevotella amni	0.897	0.607	0.659
Pseudomonas akapageensis	0.51	0.965	0.275
Ralstonia insidosa	0.285	0.014	0.45
Sneathis sp.	0.031	0.014	0.111

# DADA2 based code used for 16S sequence data analysis for ASV (Amplicon Sequence Variant) generation, related to Figure 2 and 3

Supporting links:

[https: https://github.com/benjjneb/dada2.]

[doi: 10.12688/f1000research.8986.2]

```
library("knitr")

.cran_packages <- c("ggplot2", "gridExtra")

.bioc_packages <- c("dada2", "phyloseq", "DECIPHER", "phangorn")

.inst <- .cran_packages %in% installed.packages()

if(any(!.inst)) { install.packages(.cran_packages[!.inst])}

.inst <- .bioc_packages %in% installed.packages()

if(any(!.inst)) { BiocManager::install(.bioc_packages[!.inst], ask = F)}

# Sort forward/reverse reads are in same order

fnFs <- sort(list.files(seq_path, pattern="_R1.fastq"))

fnRs <- sort(list.files(seq_path, pattern="_R2.fastq"))

# Extract sample names,given:

sampleNames <- sapply(strsplit(fnFs, "_"), `[`, 1)

fnFs <- file.path(seq_path, fnFs)

fnRs <- file.path(seq_path, fnRs)

fnFs[1:2]

fnRs[1:2]

#To check reads quality of forward and reverse reads

plotQualityProfile(fnFs[1:2])

plotQualityProfile(fnRs[1:2])

filt_path <- file.path(seq_path, "filtered_1")

if(!file_test("-d", filt_path)) dir.create(filt_path)

# this is to provide new names for the filtered .fastq files

filtFs <- file.path(filt_path, paste0(sampleNames, "_F_filt.fastq.gz"))

filtRs <- file.path(filt_path, paste0(sampleNames, "_R_filt.fastq.gz"))

out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs,

               maxN=0,
                    maxLen = c(600,600),
                    minLen = c(200,200),
                    truncQ = 0,
                    trimLeft = c(17,21),
                    compress=TRUE, multithread= TRUE,
                    verbose = TRUE)
saveRDS(out,"qaqc_out",compress = TRUE)
out<- readRDS("qaqc_out")
plotQualityProfile(filtFs[1:2])
plotQualityProfile(filtRs[1:2])
derepFs <- derepFastq(filtFs, verbose=TRUE)
names(derepFs) <- sampleNames
saveRDS(derepFs,"derepFs",compress = TRUE)
# reload the same object in R saved previously
derepFs<- readRDS("derepFs")
derepRs <- derepFastq(filtRs, verbose=TRUE)
names(derepRs) <- sampleNames
saveRDS(derepRs,"derepRs",compress = TRUE)
derepRs<- readRDS("derepRs")
errF <- learnErrors(filtFs, multithread=TRUE, verbose = TRUE)
errR <- learnErrors(filtRs, multithread=TRUE, verbose = TRUE)
#to check the error plots
plotErrors(errF)
plotErrors(errR)
dadaFs <- dada(derepFs, err=errF, multithread=TRUE)
saveRDS(dadaFs,"dadaFs",compress = TRUE)
dadaRs <- dada(derepRs, err=errR, multithread=TRUE)
saveRDS(dadaRs,"dadaRs",compress = TRUE)
dadaFs<- readRDS("dadaFs")
dadaRs <- readRDS("dadaRs")
```

```
#Now forward and reverse reads are merged to get merged sequences
mergers <- mergePairs(dadaFs, derepFs, dadaRs, derepRs, verbose = TRUE)
#A sequence table is generated to get abundance values for each sample
seqtabAll <- makeSequenceTable(mergers[names(mergers)])
table(nchar(getSequences(seqtabAll)))
seqtabNoC <- removeBimeraDenovo(seqtabAll, verbose = TRUE) # Remove chimera
dim(seqtabNoC)
sum(seqtabNoC)/sum(seqtabAll)
saveRDS(seqtabNoC,"seqtabNoC",compress = TRUE)
seqtabNoC <- readRDS("seqtabNoC")
```

# Code for diversity estimation in R tool related to Figure 2 and Figure 3

```
library("vegan")
shannon_index = diversity (file_input, index = "shannon")
simpson_simpson = diversity (file_input, index = "simpson")
```

# Core for box-plot in R tool related to Figure 2 and Figure 3

```
boxplot(data1,data2, col =c("red","green"),
      names = c("PTB","TB"), varwidth = TRUE, ylab = "Simpson index")
```

# Code used for taxonomy assignment and relative abundance estimation, related to Figure 4-Figure 10

```
######## Taxonomy annotation using reference database ##############
fastaRef <- "silva_nr_v138_train_set.fa.gz"
taxTab <- assignTaxonomy(seqtabNoC, refFasta = fastaRef, multithread=TRUE)
unname(head(taxTab))
taxa_sp <- addSpecies(taxTab, "silva_species_assignment_v138.fa.gz")
# remove OTUs unclassified on phylum level, and non bacteria
table(taxa_sp[, 1])
sum(is.na(taxa_sp[, 2]))
```

```
tmp <- taxa_sp[!is.na(taxa_sp[, 2]) & taxa_sp[, 1] == "Bacteria", ]

tax.good <- tmp[-c(grep("Chloroplast", tmp[, 4]), grep("Mitochondria", tmp[, 5])), ]

seqtab.nochim2.good <- seqtabNoC[, rownames(tax.good)]

summary(rowSums(seqtab.nochim2.good))

seqtab.nochim2.print <- t(seqtab.nochim2.good)

tax.print <- tax.good

all.equal(rownames(seqtab.nochim2.print), rownames(tax.print))

rownames(seqtab.nochim2.print) <- paste("sq", 1:ncol(seqtab.nochim2.good), sep = "")

rownames(tax.print) <- rownames(seqtab.nochim2.print)

# curate NAs in taxonomy table

Taxb <- tax.print

k <- ncol(Taxb) - 1

for (i in 2:k) {

  if (sum(is.na(Taxb[, i])) > 1) {

    test <- Taxb[is.na(Taxb[, i]), ]

    for (j in 1:nrow(test)) {

      if (sum(is.na(test[j, i:(k + 1)])) == length(test[j, i:(k + 1)])) {

        test[j, i] <- paste(test[j, (i - 1)], "_unclassified", sep = "")

        test[j, (i + 1):(k + 1)] <- test[j, i]} }

    Taxb[is.na(Taxb[, i]), ] <- test}

  if (sum(is.na(Taxb[, i])) == 1) {

    test <- Taxb[is.na(Taxb[, i]), ]

    if (sum(is.na(test[i:(k + 1)])) == length(test[i:(k + 1)])) {

      test[i] <- paste(test[(i - 1)], "_unclassified", sep = "")

      test[(i + 1):(k + 1)] <- test[i] }

    Taxb[is.na(Taxb[, i]),] <- test }}

Taxb[is.na(Taxb[, (k + 1)]), (k + 1)] <- paste(Taxb[is.na(Taxb[, (k + 1)]), k], "_unclassified", sep = "")

unname(head(Taxb))

\View(Taxb)# write output

write.table(seqtab.nochim2.print, "bac_seqtab_nochim2.txt", quote = F, sep = "\t")
write.table(Taxb, "bac_taxonomy_table.txt", sep = "\t", quote = F)

uniquesToFasta(seqtab.nochim2.good, "bac_dada2_unique_nochim.fasta")
```

# Code for bar plot in R tool related to Figure 5

```
a1 = as.matrix(data)

barplot(a1,names = colnames(A1), col= 1:5,las=2, legend = rownames(A1),ylim = c(0, 100), xlab = "x-axis name",  ylab = "y-axis name",

        xlim = c(0, ncol(a1) + 25),

        args.legend=list(

          x=ncol(a1) + 29,

          y=max(colSums(a1) +2),

          bty = "n", cex = 0.7

        ))
```

# Code for correlation plot in R tool related to Figure 8

```
library("ggpubr")

ggscatter(my_data, x = "data _1", y = "data_2", add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "Spearman", xlab = "X-axis name", ylab = "Y-axis name)")
```

# Code for correlation plot in R tool related to Figure 9 and Figure S3

```
corrplot(data.core, method="color", addCoef.col = 1,number.cex = 0.6, tl.cex = 0.8)
```

# Code used for volatility plot (qiime2 longitudinal plugin), related to Figure 10

```
qiime longitudinal plot-feature-volatility Inputs file_name --m-metadata-file metadata_file --o-visualization output_name
```

#Code for Picrust2 related to Figure 11a and 11b

```
picrust2_pipeline.py -s /home/thsti/Desktop/picrust2_out_pipeline/seq_picruST.fna -i /home/thsti/Desktop/picrust2_out_pipeline/PTB.biom -o picrust2_out_pipeline
```